Уральский федеральный университет
имени первого Президента России Б.Н.Ельцина
Уральский энергетический институт

А. Ю. КИСЕЛЬНИКОВ
П. Ю. ХУДЯКОВ
А. Ю. ЖЕРЕБЧИКОВ

ПРОГРАММИРОВАНИЕ ПТК SIEMENS И ПТК VIPA В ПРОГРАММНЫХ ПАКЕТАХ STEP7, WINCC И PCS7

Учебно-методическое пособие

Министерство образования и науки Российской Федерации
Уральский федеральный университет
имени первого Президента России Б. Н. Ельцина

А. Ю. Кисельников, П. Ю. Худяков, А. Ю. Жеребчиков

ПРОГРАММИРОВАНИЕ ПТК SIEMENS И ПТК VIPA В ПРОГРАММНЫХ ПАКЕТАХ STEP7, WINCC И PCS7

Учебно-методическое пособие

Рекомендовано методическим советом УрФУ
для студентов всех форм обучения бакалавриата
140100.62 — Теплоэнергетика и теплотехника,
15.03.04 — Автоматизация технологических процессов и производств,
13.03.01 — Теплоэнергетика и теплотехника

Екатеринбург
Издательство Уральского университета
2016

УДК 004.4(075.8)
ББК 32.973я73
К44

Рецензенты:
Э. С. Лапин — д-р техн. наук, проф., завкафедрой автоматики и комп. технологий горно-механического факультета Уральского государственного горного ун-та;
М. А. Маленцов — нач. отдела КИПиА, АСУ ТП и технологических защит Свердловского филиала ПАО «Т-Плюс»

Научный редактор — ст. препод. кафедры ТЭС *Н. А. Акифьева*

Кисельников, А. Ю.
К44 Программирование ПТК Siemens и ПТК Vipa в программных пакетах Step7, WinCC и PCS7 : учебно-методическое пособие / А. Ю. Кисельников, П. Ю. Худяков, А. Ю. Жеребчиков. — Екатеринбург : Изд-во Урал. ун-та, 2016. — 83,[1] с.
ISBN 978-5-7996-1816-2

Учебное пособие по программированию ПТК Siemens и ПТК Vipa в программных пакетах Step7, WinCC и PCS7 предназначено для студентов всех форм обучения профиля подготовки «Автоматизация технологических процессов и производств», включает содержание основных этапов работ по выполнению проекта автоматизации, методику выполнения этих этапов. Разработано с использованием материалов учебных курсов и технической документации производителей ПТК.

Библиогр.: 2 назв. Табл. 1. Рис. 96.

УДК 004.4(075.8)
ББК 32.973я73

ISBN 978-5-7996-1816-2

ОГЛАВЛЕНИЕ

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

AI	–	Analog input
AO	–	Analog output
ASCII	–	American standard code for information interchange
CPU	–	Central processing unit
DHCP	–	Dynamic Host Configuration Protocol
DI	–	Digital input
DO	–	Digital output
ES	–	Engineering station
FB	–	Function block
FC	–	Function
HMI	–	Human-machine interface
IIS	–	Internet information services
IP	–	Internet Protocol
MAC	–	Media Access Control
OS	–	Operator station
PC	–	Personal computer
PLC	–	Programmable logic controller
TCP	–	Transmission control protocol

АСУТП	–	Автоматизированная система управления технологическим процессом
ВМ	–	Виртуальная машина
ПЛК	–	Программируемый логический контроллер
ПО	–	Программное обеспечение
ПТК	–	Программно-технический комплекс
ТХК	–	Термопара хромель-копель
ЦПУ	–	Центральное процессорное устройство

ВВЕДЕНИЕ

Этап программирования ПТК является одним из наиболее важных этапов создания АСУТП технологического процесса, наряду с процессом проектирования. Наладка алгоритмов производится на последнем этапе ввода АСУТП и включает в себя огромное количество взаимосвязанных процессов.

Для ускорения процесса реализации алгоритмов управления разработчик ПТК должен обладать не только отличными знаниями аппаратной и программной части комплекса, но также иметь четкое понимание технологического процесса, его параметров и особенностей.

В данном учебно-методическом пособии авторами предпринята попытка систематизации знаний в области программирования таких ПТК, как ПТК Siemens и ПТК Vipa, имеющих достаточно высокий порог вхождения как в финансовом плане, так и в информационном.

В работе приведены основные базовые этапы конфигурирования и программирования ПТК, позволяющие неопытному пользователю начать работу в системе.

Учебное пособие по программированию ПТК Siemens и ПТК Vipa в программных пакетах Step7, WinCC и PCS7 включает содержание основных этапов работ по выполнению проекта автоматизации, методику выполнения этих этапов. Разработано с использованием материалов учебных курсов и технической документации производителей ПТК.

Предполагается, что перед освоением данного материала читатель уже имеет опыт установки и настройки операционных систем, конфигурирования виртуальных машин и базовые знания в области электротехники и электроники.

Авторский коллектив просит все замечания и предложения, касающиеся данного учебного пособия, высылать по адресу: *lumen_xp@mail.ru*.

ЧАСТЬ 1. НАСТРОЙКА ПРОГРАММНО-АППАРАТНОГО ОБЕСПЕЧЕНИЯ

1.1. Настройка виртуальной машины VMware Workstation

Для повышения удобства работы в системе и предотвращения ошибок начинающего пользователя все работы будут проводиться посредством виртуальной машины, которая уже должна иметься у слушателя.

После установки VMware Workstation, создания виртуальной машины и установки операционной системы Windows XP SP3 Professional приступаем к ее настройке (*рис. 1.1*).

Для начала необходимо настроить сеть передачи данных. Для этого заходим в меню *ВМ > Настройки*. В списке выбираем *Network adapter*, далее в *Network connection* выбираем *Bridged:...,* ставим галочку перед *Replicate* (*рис. 1.2*).

Вариант *Bridged networking* означает, что виртуальная машина будет подключаться к локальной сети, используя реальную Ethernet-плату вашего основного компьютера, которая выполняет функции «моста» между виртуальной машиной и реальной физической сетью. Это позволяет виртуальному компьютеру выглядеть со стороны реальной сети как полнофункциональный хост. Назначение сетевых адресов в этом случае осуществляется в соответствии с правилами, принятыми в реальной локальной сети. Вы можете подключаться по протоколу DHCP либо назначить статический IP-адрес.

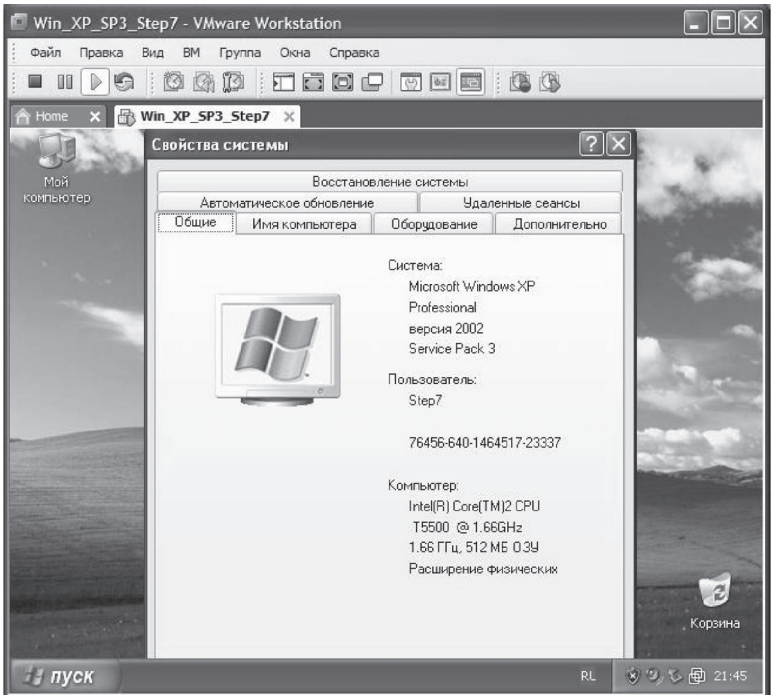


Рис. 1.1. Свойства системы

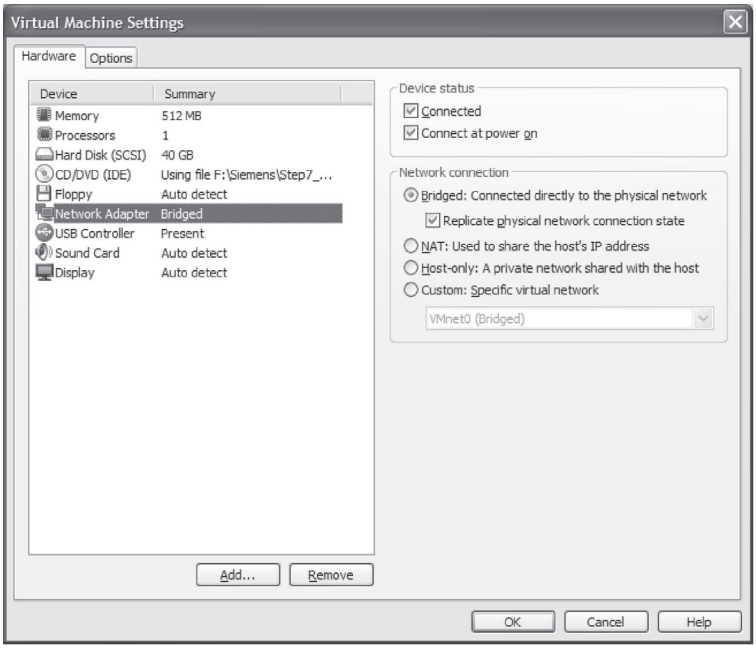


Рис. 1.2. Настройки виртуальной машины

Виртуальная машина, подключенная по этому варианту, может использовать любые сетевые сервисы, предоставляемые в локальной сети, к которой она подключена: принтеры, файл-серверы, маршрутизаторы и т. д. Точно так же она может предоставить в сеть какие-то из своих ресурсов. Это наиболее часто используемая конфигурация сетевых служб виртуального компьютера.

Следующим шагом является выбор сетевого интерфейса. Для этого необходимо зайти в меню *Правка > Управление виртуальной сетью* (*рис. 1.3*).

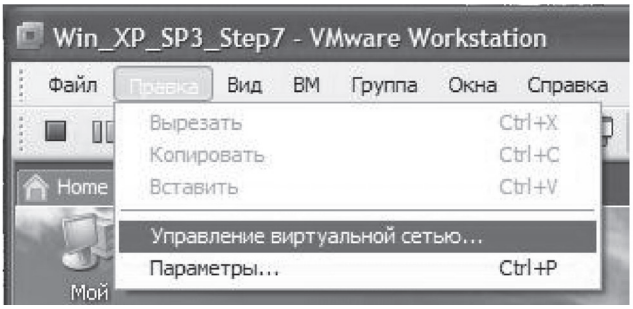


Рис. 1.3. Открытие окна управления виртуальной сетью

В окне *Virtual Network Editor* произведем дальнейшую настройку сети (*рис. 1.4*).

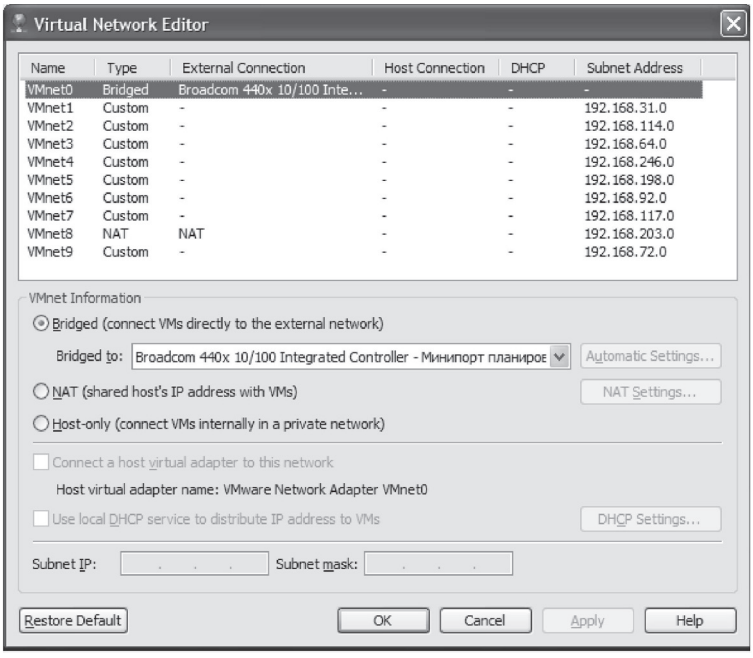


Рис. 1.4. Окно управления виртуальной сетью

За ненадобностью отключим сети VMnet1 и VMnet8, убрав галочки с *Connect a host virtual adapter to this network* и *Use local DHCP service to distribute IP address to VMs*. В настройках VMnet0 выбираем физический адаптер.

Далее переходим к настройке сети. IP-адрес шлюза 192.168.0.1, IP-адрес физического адаптера компьютера 192.168.0.2, IP-адрес виртуальной машины будет 192.168.0.3, IP-адрес контроллера позже установим как 192.168.0.5.

Для начала настроим физический адаптер, для этого перейдем в меню *Пуск* > *Панель управления* > *Сетевые подключения* и правой кнопкой нажмем на соответствующий адаптер. Выберем *Свойства* > *Протокол Интернета (TCP/IP)* и произведем настройку следующим образом (*рис. 1.5*).

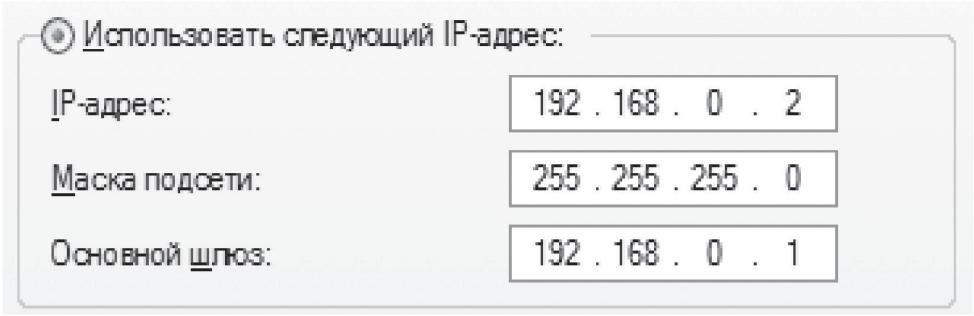


Рис. 1.5. Назначение IP адреса физического адаптера компьютера

Соответственно на виртуальной машине выставим следующие настройки (*рис. 1.6*).

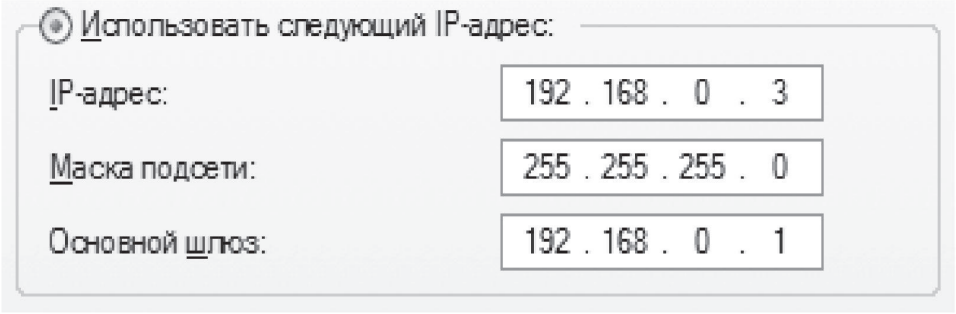


Рис. 1.6. Назначение IP-адреса виртуальной машины

Проверить наличие устойчивого соединения можно в командной строке Windows командой ping. На виртуальной машине пройдем в меню *Пуск > Все программы > Стандартные > Командная строка* и выполним команду ping 192.168.0.2. Полученный результат отображен на *рис. 1.7.*

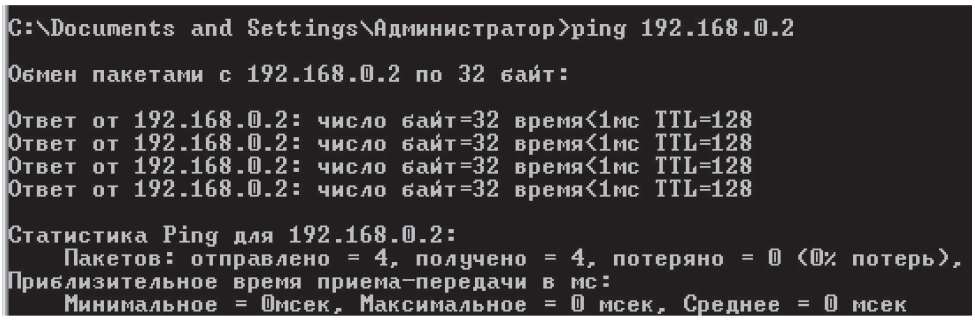


Рис. 1.7. Окно командной строки

На этом настройка виртуальной машины заканчивается. Приступаем к установке программного обеспечения Siemens Simatic.

1.2. Установка программного обеспечения

Перечень программного обеспечения, устанавливаемого на виртуальной машине:

- Step7 5.5;
- WinCC 7.0 SP1 HF1 + Русификатор;
- Microsoft Excel 2007;
- Компоненты Windows: Internet Information Services (IIS) и Очереди сообщений.

Начнем с установки Step7 5.5 из релиза Step7 Professional 2010.

При установке выбираем необходимые компоненты и в конце установки соглашаемся с лицензионным соглашением (*рис. 1.8*).

Перед установкой WinCC необходимо установить компоненты Windows, для этого нам понадобится установочный диск с Windows XP. Зайдем в меню *Пуск > Панель управления > Установка и удаление программ > Установка компонентов Windows.* Выбираем *Internet Information Services (IIS)* и *Очереди сообщений* (*рис. 1.9*).

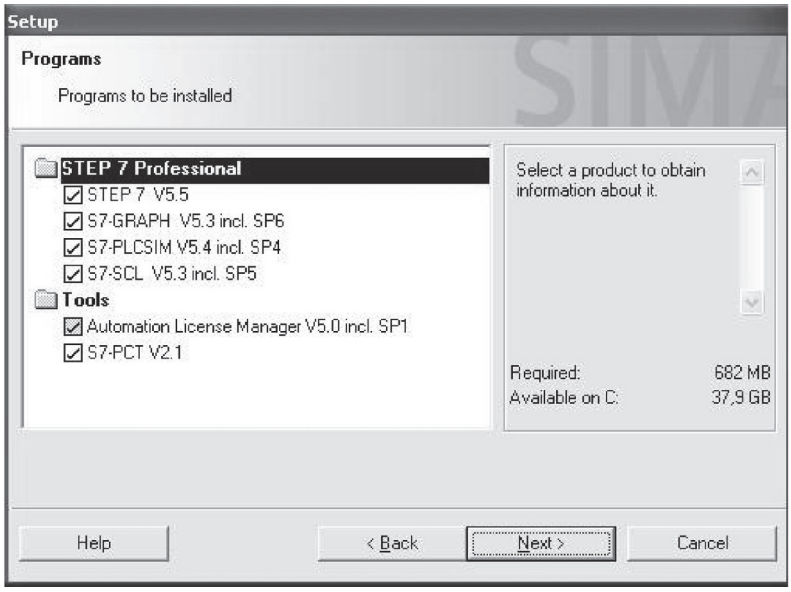
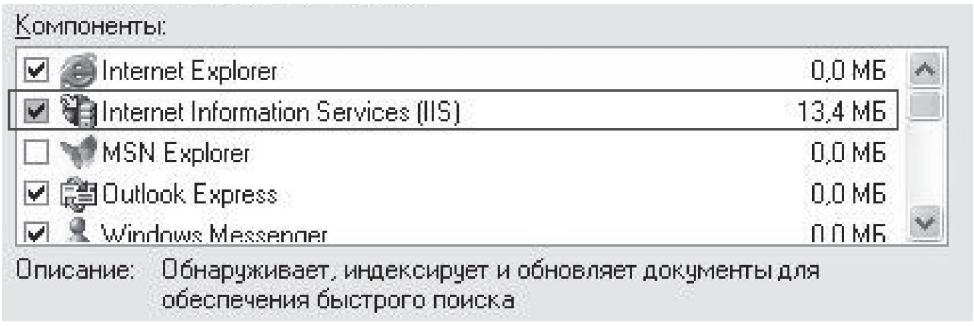


Рис. 1.8. Установка программного обеспечения Step7



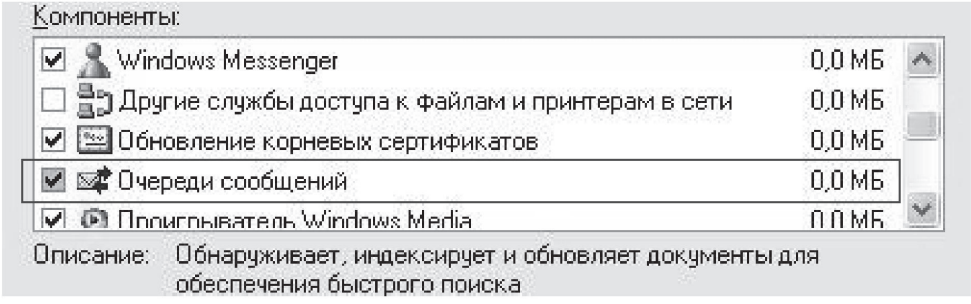


Рис. 1.9. Установка компонентов Windows

Нажимаем *OK*, система запросит установочный диск, с которого и будут установлены компоненты.

Установка Microsoft Excel 2007 происходит обычным образом, и на ней подробно останавливаться не будем. Данное программное обеспечение необходимо для ряда компонентов WinCC.

Во время установки WinCC выбираем *User-defined installation*, далее выбираем *WinCC Expert mode* (*рис. 1.10*).

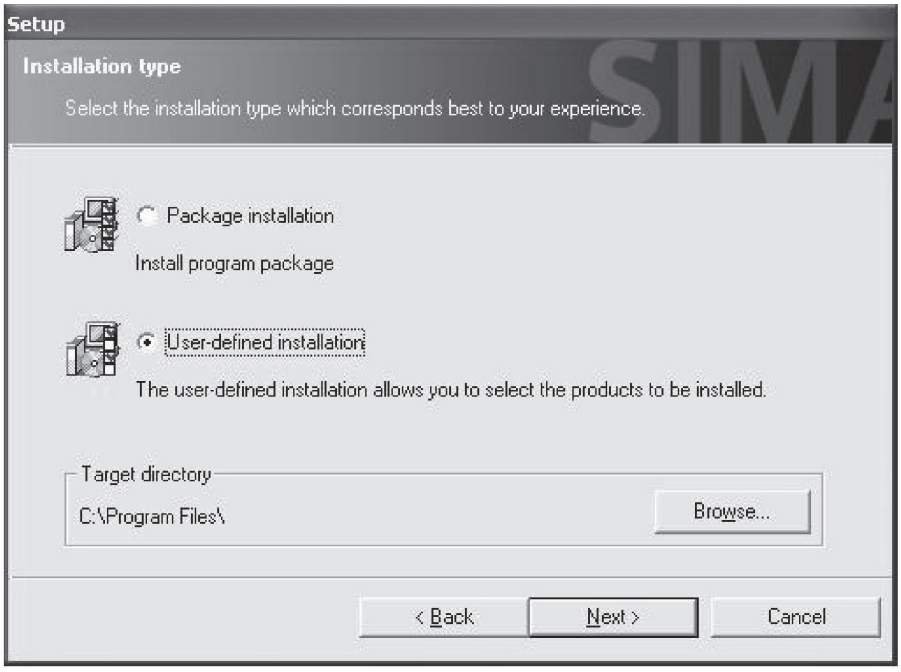


Рис. 1.10. Выбор режима установки WinCC — User-defined installation

После выбора данного варианта появляется возможность выбора компонентов для установки. Если позволяет дисковое пространство и имеющиеся лицензии, то советуем выбрать максимально возможный набор программ (*рис. 1.11*).

Достаточно важным компонентом является AS-OS Engineering. Данный компонент позволит использовать в WinCC таблицу символов из проекта Step7 (*рис. 1.12*).

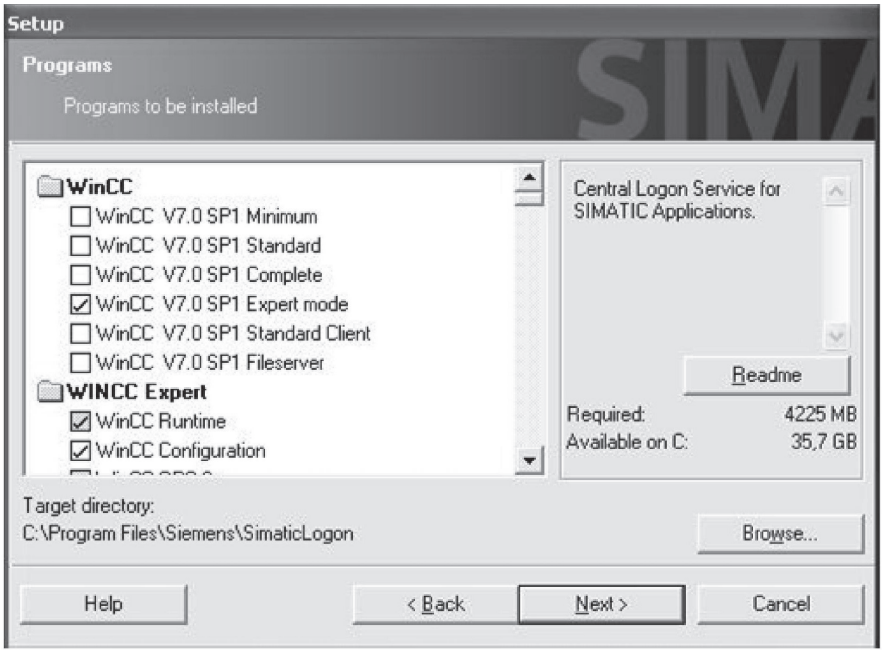


Рис. 1.11. Выбор максимально возможного набора программ

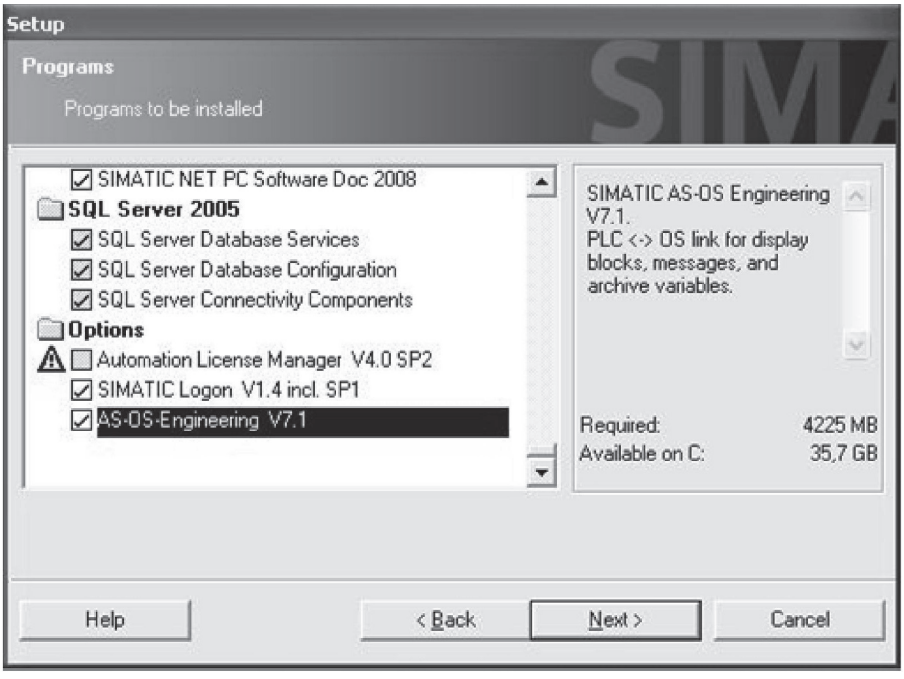


Рис. 1.12. Выбор компонента AS-OS Engineering

После установки WinCC необходимо установить последние обновления и русификацию (если это необходимо).

Следующим этапом является установка ключей программного обеспечения. Эта процедура производится любым из доступных вам способов. Данный этап является завершающим при установке программного обеспечения.

После выполнения этих действий вы будете иметь виртуальную машину с комплектом ПО, необходимым для работы с контроллерами Siemens и VIPA.

1.3. Сброс памяти ЦПУ

Перед началом создания нового проекта необходимо очистить память ЦПУ от старого проекта. Различают два вида очистки памяти:

1) Overall Reset — общий сброс. Выполняется очистка памяти от пользовательских программ, при этом все системные файлы и конфигурация оборудования не затрагиваются;

2) Factory Reset — восстановление заводских настроек. Выполняется полная очистка памяти до заводских настроек, при этом удаляется конфигурация оборудования и сбрасываются настройки IP-адреса на 0.0.0.0 и MPI-адреса на 2.

Для совершения общего сброса необходимо выполнить ряд действий с переключателем режимов ЦПУ, изображенный на *рис. 1.13*:

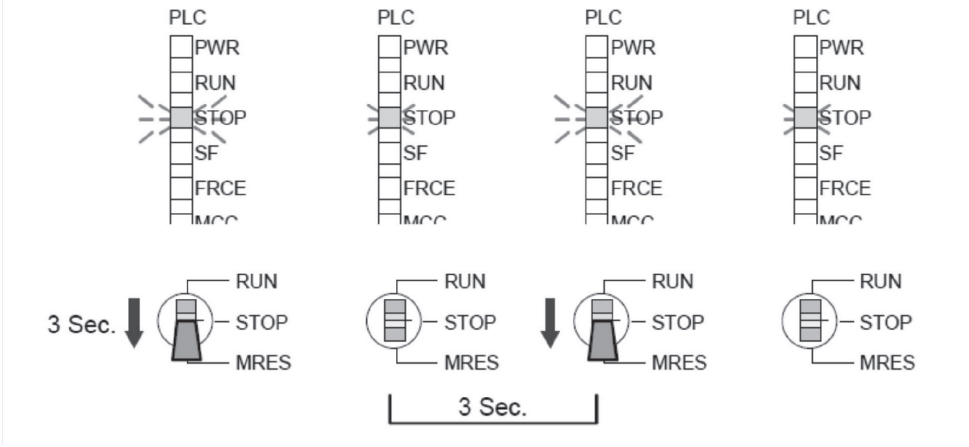


Рис. 1.13. Общий сброс памяти ЦПУ с помощью переключателя режимов работы ЦПУ

- перевести переключатель в положение MRES и удерживать его в этом положении около 3 с;
- после того как светодиод STOP перестанет мигать и начнет постоянно гореть, необходимо перевести переключатель в положение STOP, затем сразу в положение MRES и быстро вернуть его в положение STOP (в течение не более 3 с);
- после данных действий светодиод STOP начнет быстро мигать и после очистки памяти загорится ровным светом.

Для восстановления заводских настроек необходимо выполнить ряд действий с переключателем режимов ЦПУ, изображенный на *рис. 1.14*:

- перевести переключатель в положение MRES и удерживать его в этом положении около 30 с. В момент перевода переключателя в положение MRES светодиод STOP начинает мигать несколько секунд, затем несколько секунд будет светиться ровным светом;
- после того как произойдет 6-кратное изменение режима светодиода STOP с мигания на постоянное свечение, необходимо перевести переключатель в положение STOP, затем сразу в положение MRES и быстро вернуть его в положение STOP (в течение 1 с);
- после данных действий один раз загорается светодиод. Это сигнализирует, что память полностью очищена;
- для подтверждения выполнения процедуры полной очистки памяти загораются светодиоды PWR, STOP, SF, FRCE и MCC (если этого не произошло, то восстановление заводских настроек не удалось, выполнился общий сброс, и необходимо повторить данную процедуру);
- в конце процедуры восстановления заводских настроек, если статически горят светодиоды PWR, STOP, SF, FRCE и MCC, необходимо отключить и затем включить блок питания.

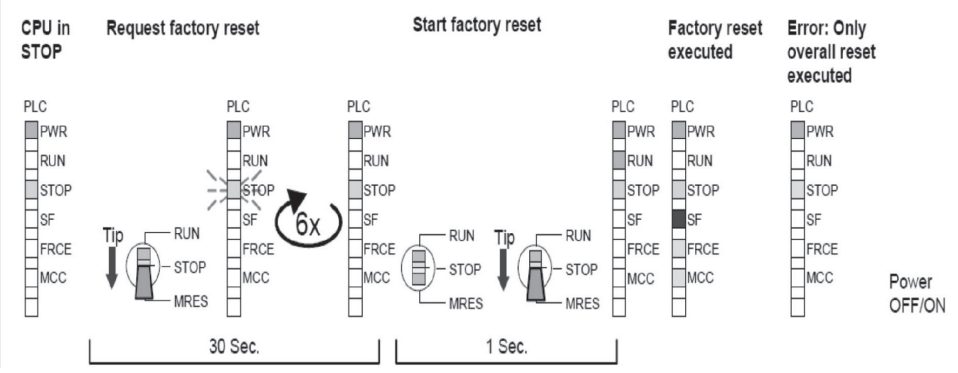


Рис. 1.14. Восстановление заводских настроек ЦПУ с помощью переключателя режимов работы ЦПУ

1.4. Состав аппаратной части

В проекте будет использоваться контроллер VIPA CPU 314SC/DPM, заказной номер 314–6CG03. Кратко о характеристиках контроллера: модуль CPU 314SC/DPM, рабочая память 128 кБ (расширение до 1 МБ), MMC, MPI, Ethernet (PG/OP), RS-485: Profibus DP или PtP (ASCII, STX/ETX, 3964R, Modbus, USS), DI 24x24 В/DO 16x24 В/0,5 А, AI 4x12 бит/AO 2x12 бит/AI 1xPt100, 4 счетчика. Программирование контроллеров VIPA серии 300 осуществляется аналогично контроллерам Siemens S7–300. Все действия будут описаны для контроллера VIPA, но они идентичны при работе с Siemens.

Модуль ввода аналоговых сигналов SM 331 AI8x12Bit, заказной номер 6ES7 331–7KF01–0AB0.

В качестве операторской и инженерной станции будет использован IBM совместный PC.

Для создания проекта необходимо запустить SIMATIC MANAGER. После открытия необходимо создать новый проект, выбрав *File* > *New*, назначить его имя и выбрать папку сохранения. После создания нового проекта выполняется команда *Insert* > *Station* > *Simatic 300 station* (*рис. 1.15*).

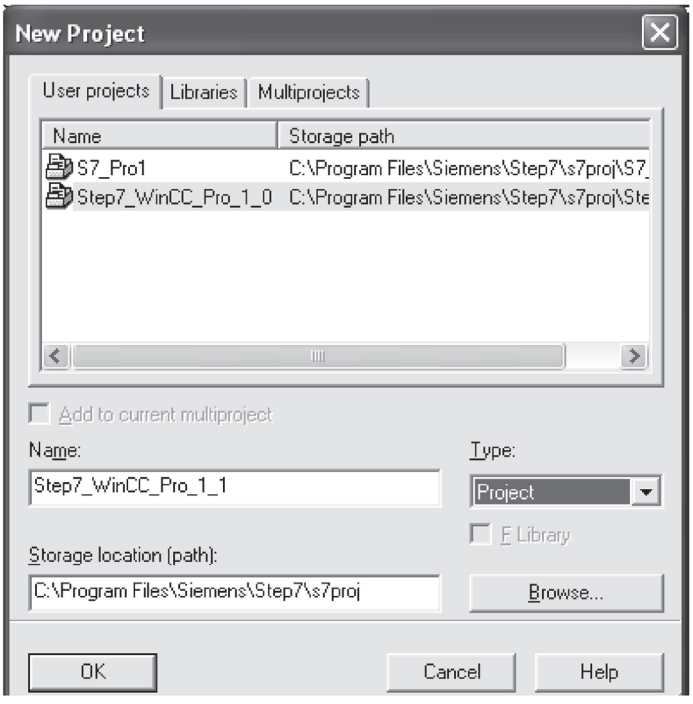


Рис. 1.15. Создание нового проекта

Для создания аппаратной конфигурации необходимо вызвать подпрограмму *Hardware configuration* (*рис. 1.16*). Выбор оборудования ПЛК подробно описан в подразделе 1.6.

В конечном итоге окно *Hardware Configuration* должно выглядеть аналогично изображенному на *рис. 1.16.* Нажимаем на кнопку *Save and Compile*.

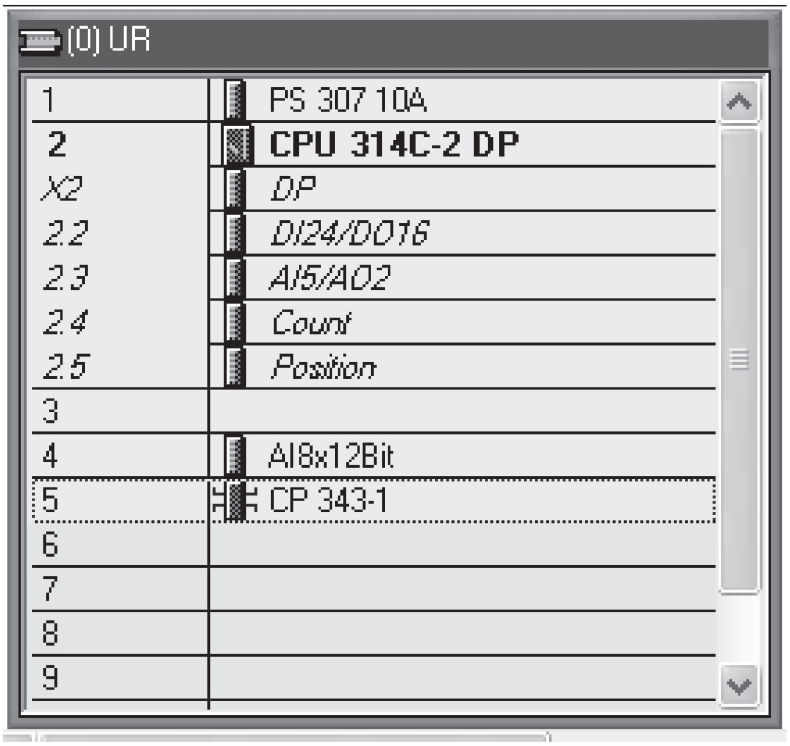


Рис. 1.16. Комплектация контроллера в редакторе Hardware

1.5. Настройка канала связи между PC и CPU

Как уже говорилось ранее, программирование контроллера VIPA происходит с помощью программного пакета Step7. Для создания проекта необходимо запустить SIMATIC MANAGER. После открытия необходимо создать новый проект, выбрав *File* > *New*, назначить его имя и выбрать папку сохранения, как показано на *рис. 1.17.*

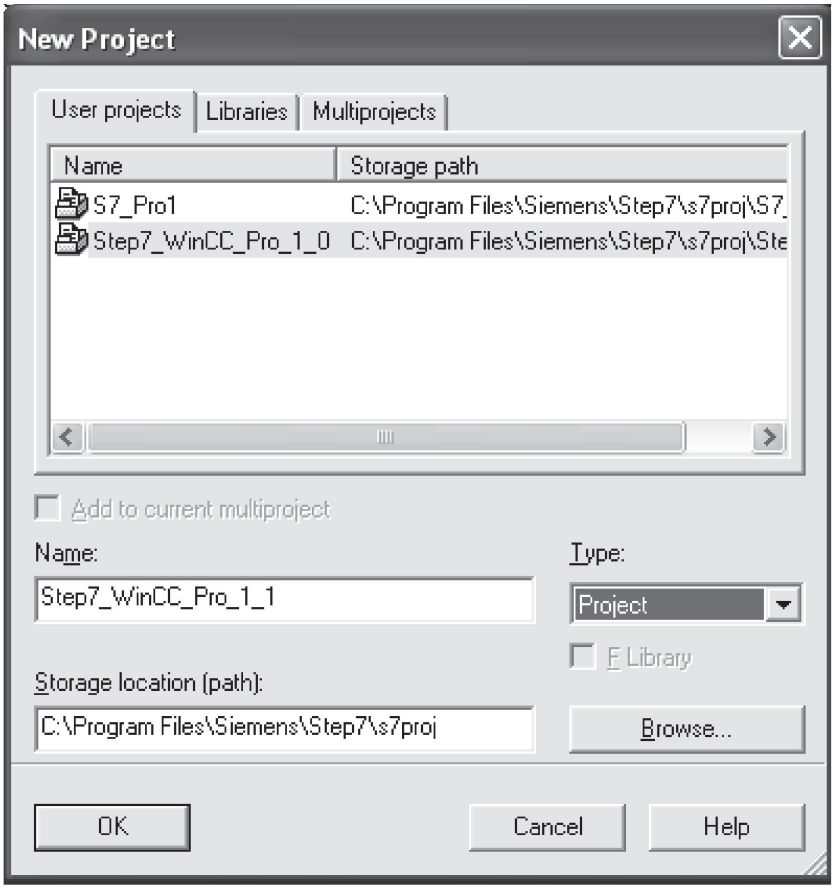


Рис. 1.17. Окно создания нового проекта

Теперь необходимо настроить соединение по сети между программатором и ПЛК.

В качестве коммуникационного кабеля был выбран наиболее доступный вариант — использование Ethernet-интерфейса. Этот вариант привлекателен тем, что, обеспечивая надежность и высокую скорость передачи данных, он не требует приобретения специальных кабелей с преобразователями интерфейса RS232-MPI, а требует лишь наличия обыкновенной витой пары.

Контроллер VIPA имеет встроенный коммуникационный процессор CP-343. Для создания подключения между программатором и ПЛК необходимо соединить их Ethernet-кабелем напрямую (CROSS) либо через маршрутизатор. После выполнения данной процедуры настраиваем сетевое подключение в компьютере, задав в свойствах сетевых

подключений IP-адрес компьютера (в нашем случае был назначен адрес 192.168.0.3) и маску подсети (в нашем случае 255.255.255.0), как было показано в п. 1.1.

Затем в SIMATIC MANAGER в меню *Option* выбираем команду *Set PG/PC Interface*. Открывается окно, изображенное на *рис. 1.18.* Необходимо установить настройки, изображенные на нем.

Если все сделано правильно, то при нажатии на кнопку *Accessible Nodes* (Доступные узлы) откроется окно, изображенное на *рис. 1.19.* В окне отображается один доступный узел по сети Industrial Ethernet. Данным узлом является контроллер VIPA, который пока определяется только по MAC-адресу, назначенному устройству на заводе-изготовителе.

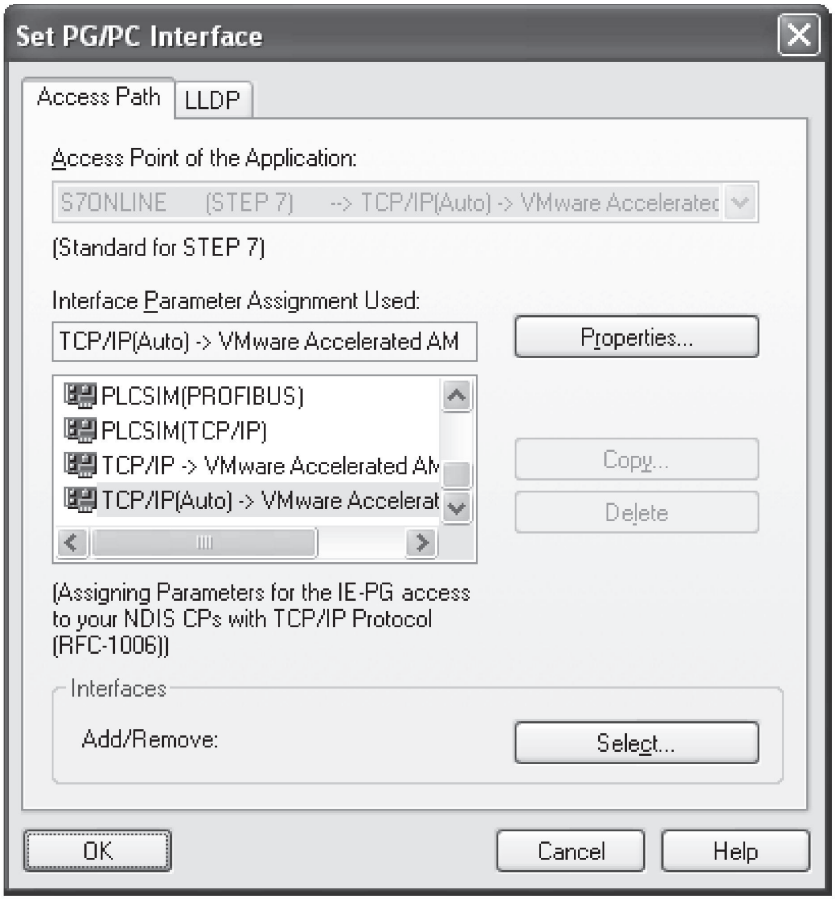


Рис. 1.18. Настройка интерфейса связи между программатором и ПЛК в окне Set PG/PC Interface

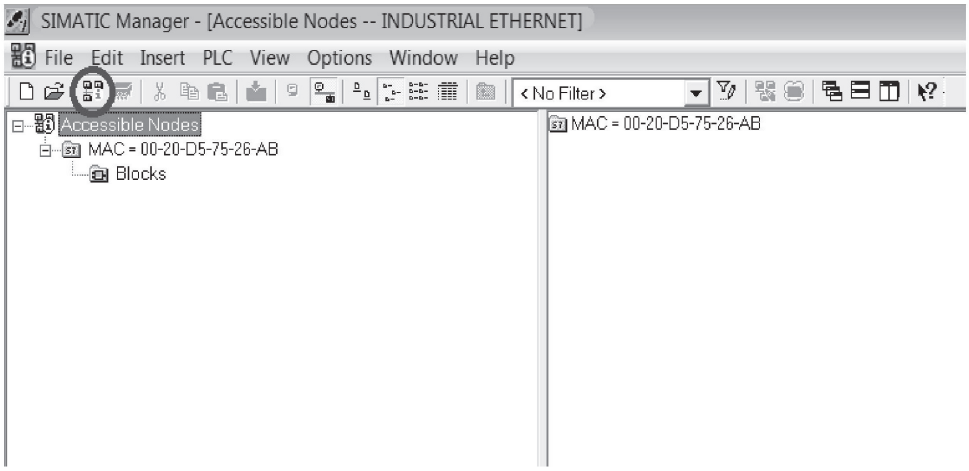


Рис. 1.19. Доступные узлы

Затем следует осуществить конфигурацию сетевых параметров ПЛК VIPA. Для этого необходимо в Simatic Manager выполнить следующие операции согласно *рис. 1.20*:

- в меню *PLC* выбрать команду *Edit ethernet nodes*;
- в поле *MAC adress* ввести MAC-адрес коммуникационного процессора CP-343 либо, нажав на *Browse* в появившемся списке, найти необходимый и выбрать его;
- в поле *IP address* назначить IP-адрес для контроллера (в нашем случае был назначен 192.168.0.5);
- в поле *Subnet mask* назначить маску подсети, такую же, как и для компьютера;
- подтвердить настройку кнопкой *Assign IP configuration*.

Проверить наличие устойчивого соединения можно в командной строке WINDOWS командой *ping <ip address>*, где <ip address> — назначенный ПЛК IP-адрес.

MAC-адрес — уникальный шестнадцатеричный серийный номер, присваиваемый каждому устройству Ethernet для идентификации его в сети; для контроллера VIPA он указан на стенке коммуникационного отсека.

На этом конфигурация соединения между программатором и ПЛК закончена. Теперь существует возможность обмена данными между ПЛК VIPA и компьютером по сети Ethernet.

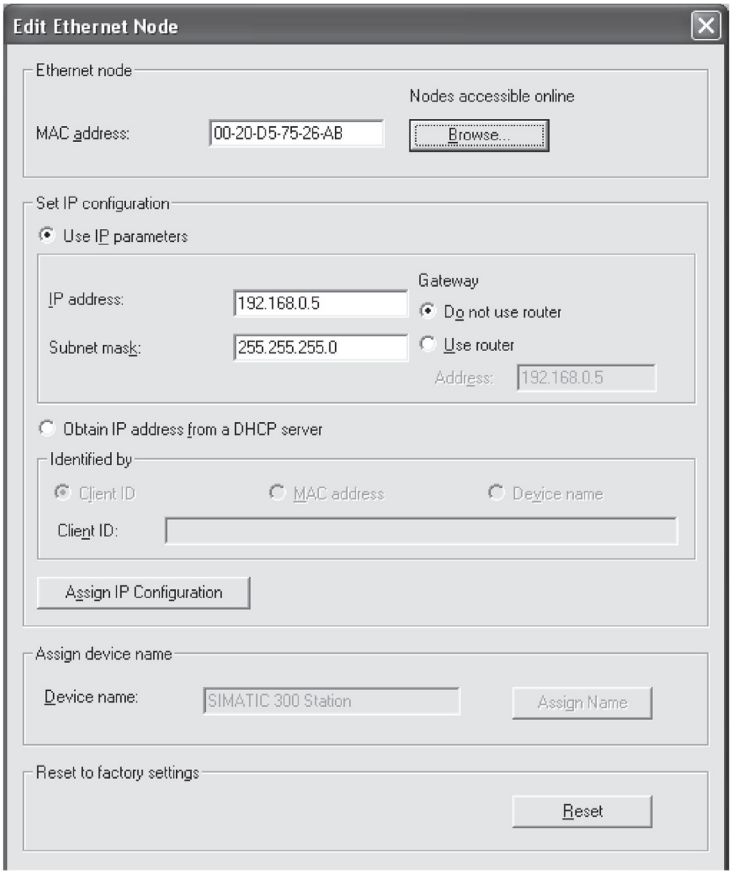


Рис. 1.20. Редактирование Ethernet-узла

1.6. Конфигурация аппаратной части

После того как связь по Ethernet с контроллером настроена, можно переходить к конфигурации оборудования.

Для этого можно использовать два пути:

1) создать конфигурацию оборудования самим в *Hardware configuration* и загрузить ее в контроллер;

2) выгрузить конфигурацию оборудования из контроллера, которую он сам создаст автоматически.

В случае конфигурирования по первому варианту необходимо выполнить команду в Simatic Manager *Insert > Station > Simatic 300 station*. Затем два раза кликнуть на элементе *Hardware*, расположение которого показано на *рис. 1.21.*

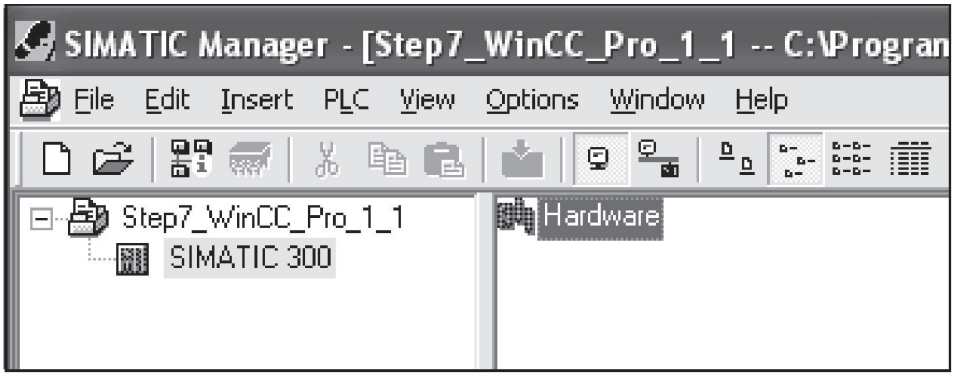


Рис. 1.21. Расположение элемента Hardware в дереве проекта

Открывается подпрограмма *Hardware configuration*, в которой и производится аппаратная конфигурация оборудования. В открывшемся окне необходимо выбрать из каталога следующие элементы:

- *Simatic 300 > Rack 300 > Rail* — конфигурационная рейка для создания аппаратной конфигурации системы;
- *Simatic 300 > PS — 300 > PS 307 10A* — блок питания (6ES7 307–1KA01–0AA0, Load supply voltage 120/230 VAC:24 VDC/10 A);
- *Simatic 300 > CPU 300 > CPU314C-2 DP > 6ES7 314–6CG03–0AB0* — ПЛК Simatic (версия прошивки 2.0), являющийся аналогом контроллера VIPA CPU314SC и используемый нами для создания проекта, при этом для контроллера на рейке однозначно закреплен слот № 2;
- *Simatic 300 > SM 300 > AI 300 > SM 331 AI8x12Bit 6ES7 331–7KF01–0AB0* — модуль ввода аналоговых сигналов, выступающий в нашем случае аналогом модуля расширения VIPA SM 331;
- *Simatic 300 > CP 300 > Industrial Ethernet > CP 343–1 > 6GK7 343–1EX11–0XE0* — модуль расширения контроллеров Simatic, выступающий в создаваемом проекте в качестве Ethernet-порта контроллера VIPA CPU 314SC, при этом на рейке этот модуль должен располагаться последним в списке (это обусловлено требованиями взаимозаменяемости контроллеров VIPA и Simatic).

В конечном итоге окно *Hardware Configuration* должно выглядеть аналогично изображенному на *рис. 1.22.*

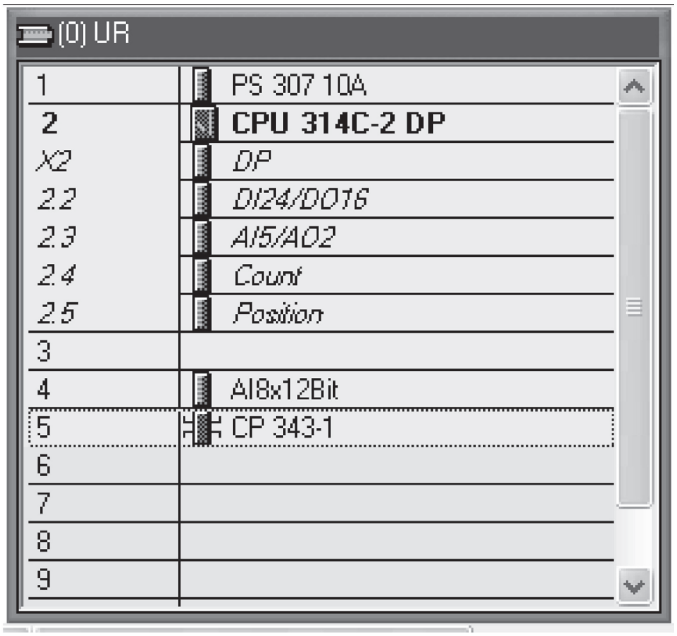


Рис. 1.22. Hardware Configuration после создания конфигурации оборудования

Нажимаем на кнопку *Save and Compile* и загружаем эту конфигурацию в контроллер с помощью кнопки *Download*. При загрузке аппаратной части в контроллер он автоматически перейдет в *Stop*, после чего появится диалоговое окно, в котором можно снова перевести его в *Start*. Теперь конфигурация оборудования настроена и загружена в контроллер.

Существует более простой способ конфигурирования оборудования. Для его использования в меню *PLC* выбираем команду Upload Station to PG. В открывшемся окне нажимаем кнопку View. На *рис. 1.23* показано, как в поле *Accessible Nodes* –Доступные узлы — был найден наш контроллер. Выбираем его и нажимаем кнопку *OK*.

После этого происходит выгрузка аппаратной конфигурации, и она автоматически появляется в проекте, как показано на *рис. 1.24*.

Теперь открываем *Hardware Configuration* и смотрим, какую конфигурацию оборудования мы получили. На *рис. 1.25* видно, что в поле аналогового модуля стоит знак вопроса. Это означает, что контроллер не смог определить точные его характеристики. Кроме того, отсутствует блок питания в первом слоте и коммуникационный процессор CP 343–1 в пятом слоте.

Блок питания для контроллеров трехсотой серии не имеет значения при конфигурации оборудования, поэтому первый слот можно оставить пустым. Но поскольку нам известна модель блока питания, то мы заполним первый слот, как описано выше в первом способе конфигурации оборудования.

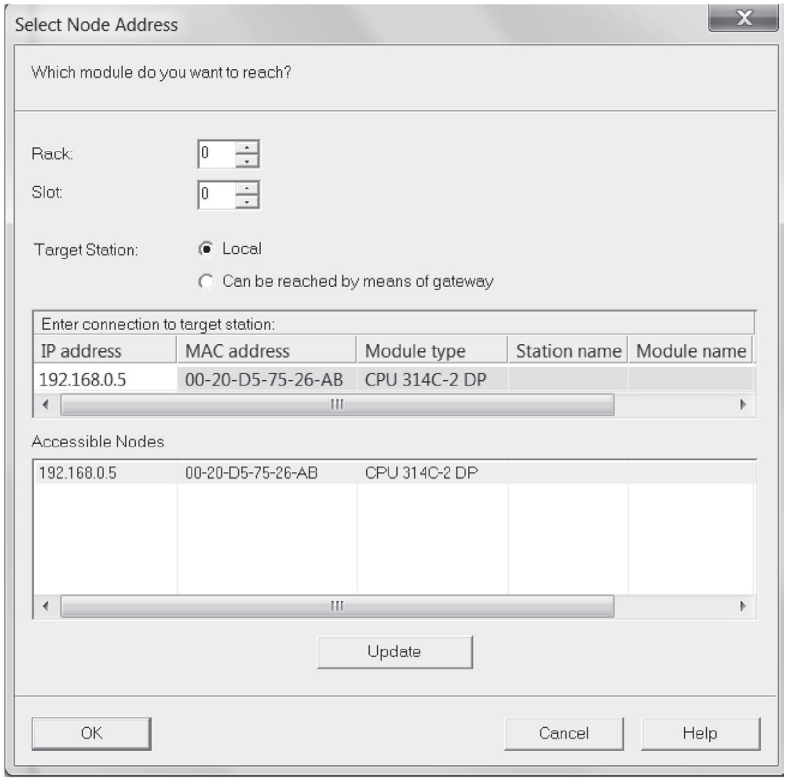


Рис. 1.23. Выбор адреса контроллера, из которого будет выполнена выгрузка аппаратной конфигурации оборудования

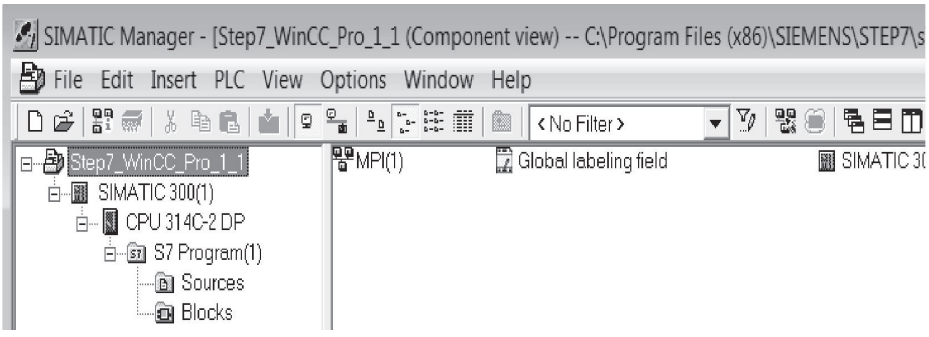


Рис. 1.24. Автоматическое создание в дереве проекта станции SIMATIC 300 (1) после выгрузки конфигурации оборудования из контроллера

На слот 4, содержащий модуль аналогового ввода, нужно кликнуть два раза и в открывшемся окне выбрать требуемый модуль.

Коммуникационный процессор CP 343–1 в 5 слот нужно добавить таким же образом, как описано выше в первом способе конфигурации оборудования.

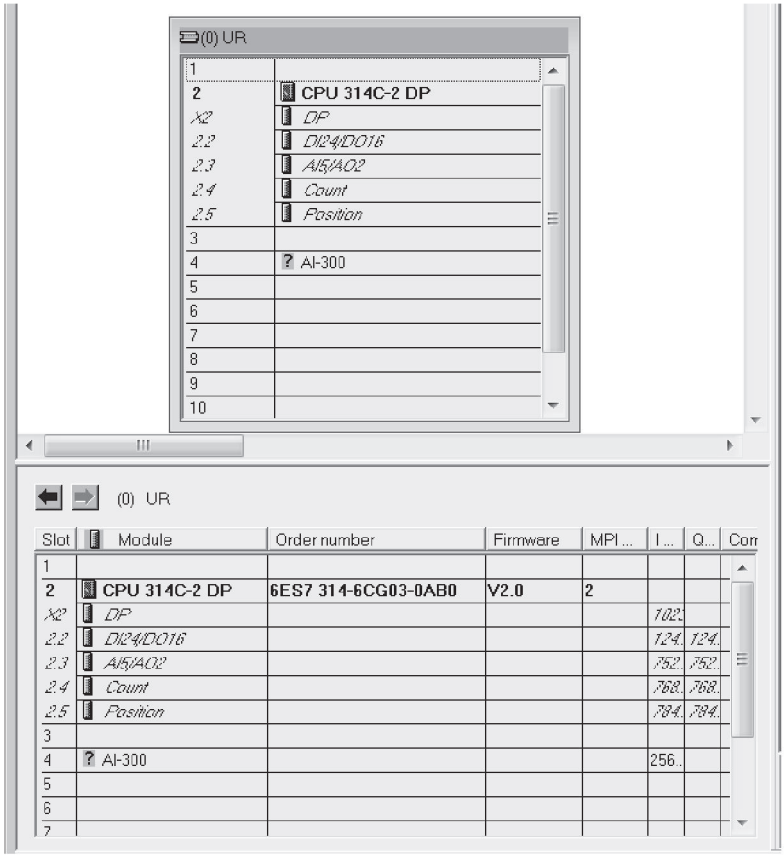


Рис. 1.25. Hardware Configuration после выгрузки конфигурации оборудования из контроллера

По *рис. 1.22* и *1.25* можно заметить, что 3 слот остался пустым, хотя в реальной стойке мы не оставили пустого места. Это обусловлено тем, что 3 слот программно зарезервирован за интерфейсным модулем IM-300, который всегда устанавливается за ЦПУ. Поскольку мы не имеем данного модуля, то оставляем этот слот пустым.

Нажимаем на кнопку *Save and Compile* и загружаем эту конфигурацию в контроллер с помощью кнопки *Download*. Не забываем, что при

загрузке аппаратной части в контроллер он автоматически перейдет в *Stop*, после чего появится диалоговое окно, в котором можно снова перевести его в *Start*. Теперь конфигурация оборудования настроена и загружена в контроллер с помощью второго способа конфигурации.

Второй способ конфигурации оборудования удобен при работе с большими проектами, в которых присутствует распределенная периферия, такая как стойки ET-200 и т. д. В этом случае программист не создает конфигурацию вручную, в соответствии с данными проекта, а лишь корректирует выгруженную из контроллера карту оборудования.

1.7. Конфигурирование рабочей станции

Для того чтобы добавить рабочую станцию, необходимо выполнить *Insert > Station > Simatic PC Station*.

Конфигурирование станции производится через HW Config. Необходимо выбрать из каталога следующие элементы:

- *Simatic PC Station > HMI > WinCC Appl.* — приложение для проектов WinCC;
- *Simatic PC Station > CP Industrial Ethernet > IE General > SW V 6.2* — замена прочих интерфейсов связи, включая TCP/IP.

Для создания сети передачи данных запустите *NetPro*, нажав на кнопку *Configure Network*.

Добавьте из библиотеки сеть Industrial Ethernet (Subnets).

Подключите коммуникационный процессор контроллера и рабочую станцию к сети передачи данных. Для этого двойным кликом мыши на соответствующем модуле зайдите в свойства, откроются окна, показанные на *рис. 1.26* и *1.27*. Нажмите на кнопку *Properties*, пропишите IP-адрес и маску подсети, указанные ранее. В разделе *Subnets* выберите *Ethernet (1)*.

После окончания настройки окно *NetPro* должно выглядеть так, как показано на *рис. 1.28*. Нажмите на *Save and Compile*.

Перейдите в окно *Simatic Manager*, выбрав в списке *SIMATIC 300 Station*, нажмите на кнопку *Download*. При этом в контроллер загрузится не только конфигурация аппаратной части проекта, но и пока еще пустой проект. На этом конфигурирование контроллера заканчивается.

Продолжим настройку рабочей станции Simatic Manager. Правой кнопкой нажмите на рабочую станцию *OS (1)* и выберите *Open Object*. Откроется окно проводника WinCC.

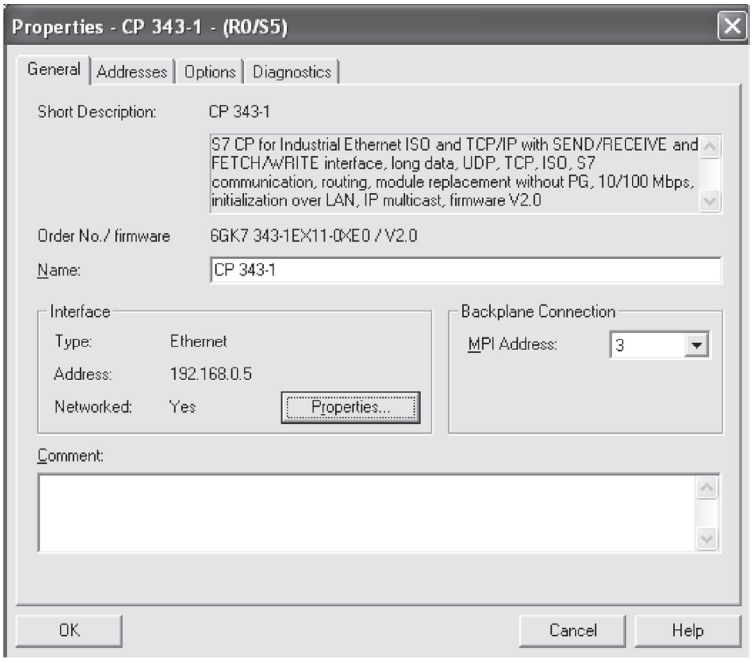


Рис. 1.26. Подключение интерфейсного модуля контроллера к сети передачи данных

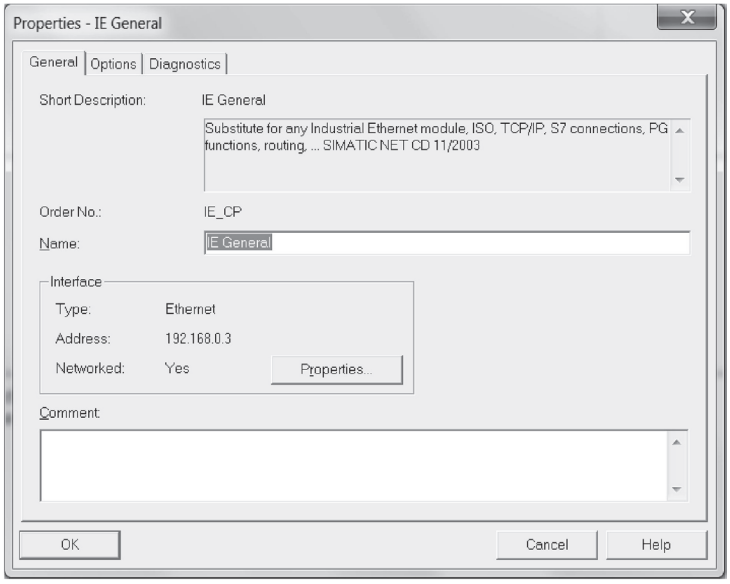


Рис. 1.27. Подключение рабочей станции к сети передачи данных

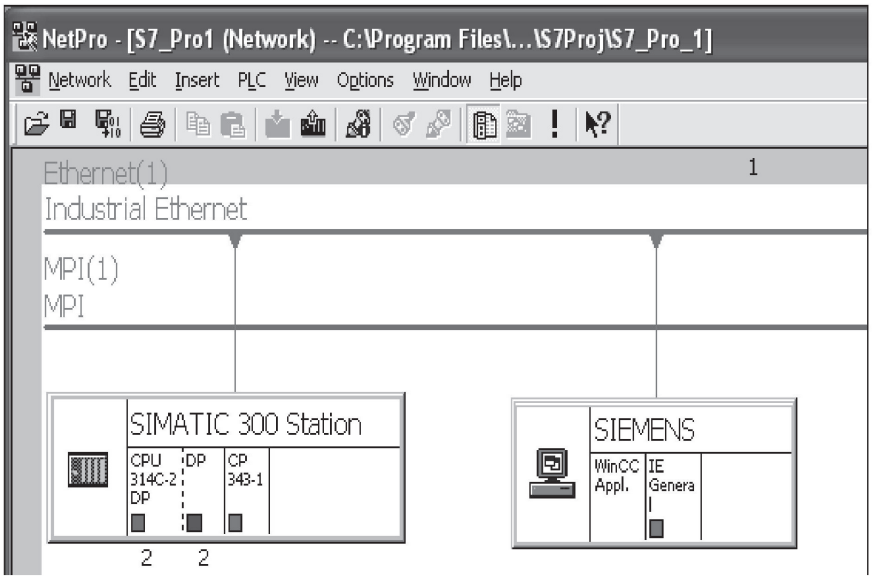


Рис. 1.28. Создание сети передачи данных Ethernet (1) в NetPro

Для установки связи с контроллером необходимо добавить интерфейсы. Нажмите на пункт *Управление тегами* правой кнопкой и выберите *Добавить драйвер*, как показано на *рис. 1.29.*

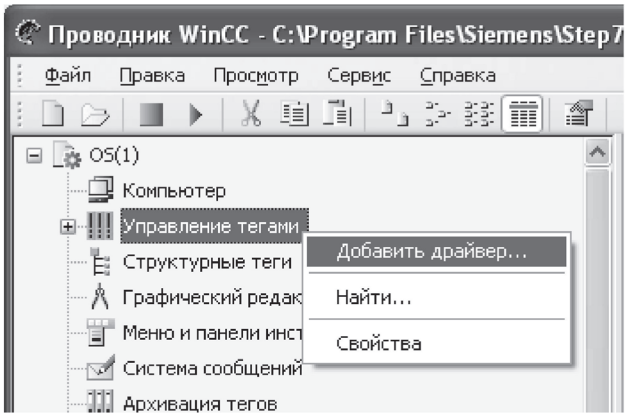


Рис. 1.29. Добавление интерфейса связи в Проводнике WinCC

Из списка выберите *SIMATIC S7 Protocol Suite*. В результате чего будет добавлен интересующий нас *TCP/IP*. В пункте *TCP/IP* создаем новое соединение, в свойствах указываем IP-адрес контроллера, номер корзины 0, номер слота 2. Именно во второй слот рейки при конфигурировании мы установили контроллер.

Правой кнопкой нажимаем на *TCP/IP* и выбираем пункт *Системные параметры* (*рис. 1.30*).

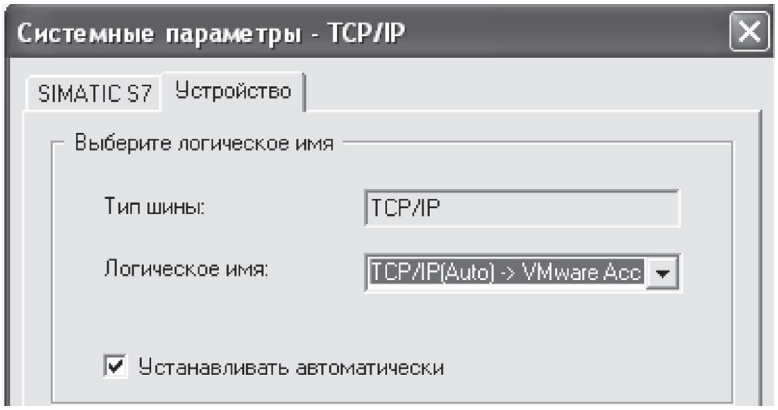


Рис. 1.30. Настройка протокола TCP/IP

Во вкладке *Устройство* выбираем *TCP/IP (Auto) > VMware*.

Для проверки соединения с контроллером нажимаем на кнопку *Запустить*. После запуска открываем меню *Сервис > Состояние связи с устройствами*. Если все было выполнено верно, вы должны увидеть окно, представленное на *рис. 1.31.*

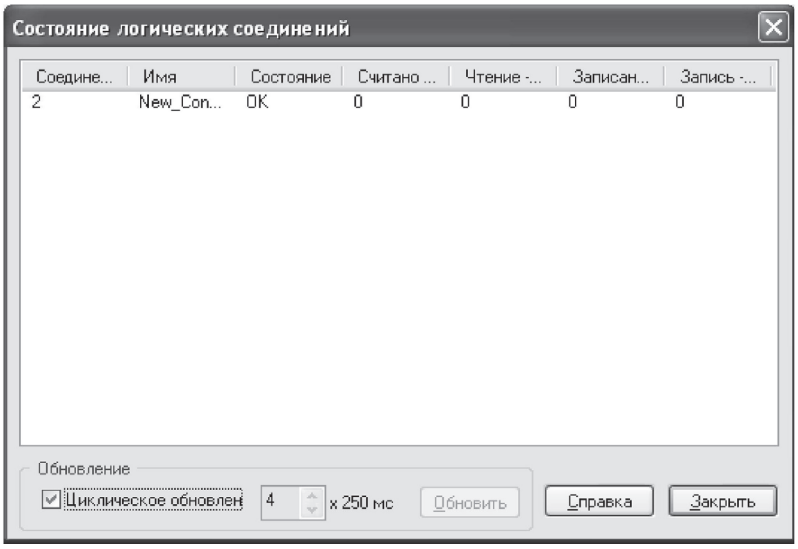


Рис. 1.31. Состояние логических соединений

На этом конфигурирование аппаратной части заканчивается, приступаем к программированию.

ЧАСТЬ 2. ПРОГРАММИРОВАНИЕ В СРЕДЕ STEP7

2.1. Последовательность работы

Работа со средой Step7 начинается после получения следующих исходных данных от Проектировщика:

1) схема электрическая принципиальная;
2) схемы электрические подключения датчиков (без клеммников шкафов ПТК);
3) алгоритмы управления;
4) план расположения оборудования.

Принципиальная схема и алгоритмы необходимы для определения логики работы конкретных исполнительных механизмов и системы в целом; основываясь на них, мы будем разрабатывать программные блоки.

Схемы подключения датчиков и планы расположения необходимы для того, чтобы сконфигурировать оборудование (в проекте может быть несколько шкафов с ПЛК, организованных по территориальному принципу) и сформировать таблицу символов.

В данном пособии мы рассматриваем совместную работу программ Step7 и WinCC на примере трех разных задач:

– отображение аналоговых сигналов при помощи датчика давления;
– отображение дискретных сигналов о положении задвижки;
– управление задвижкой при помощи дискретных сигналов.

2.2. Таблица символов

Таблица содержит символьные имена ячеек памяти области входов-выходов (адресов) ПЛК, ячеек меркерной (область памяти ПЛК, используемая для хранения данных) области памяти, блоков данных, задействованных в проекте.

Для нас эта таблица интересна прежде всего тем, что каждому определенному адресу (например, IW 124) можно присвоить имя (например, «Давление воды на входе в насос») и при интеграции программы в контроллер через организационный блок оперировать не обезличенными цифрами, а технологически осмысленными терминами.

Кроме того, в процессе разработки проекта некоторые адреса ПЛК могут сменить свой номер, данное изменение отразится в таблице символов и автоматически пропишется в организационном блоке. Если таблицей не пользоваться, то все изменения придется прописывать вручную.

Порядок работы следующий: на основе планов расположения оборудования формируются шкафы автоматики в приложении «Hardware» (см. «Создание проекта в аппаратной части») по территориальному принципу и определяется необходимое количество модулей ввода-вывода. Задаем (или фиксируем) адреса каналов в этих модулях, после чего прописываем соответствующие адреса в таблице символов (для удобства работы каждому адресу присваиваем имя).

Таблицу символов можно открыть в Simatic Manager в S7Program, щелкнув два раза на *Symbols* (*рис. 2.1*).

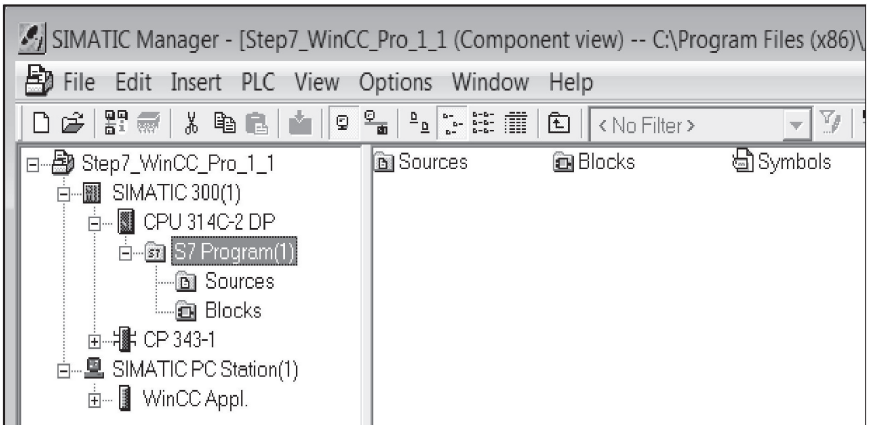
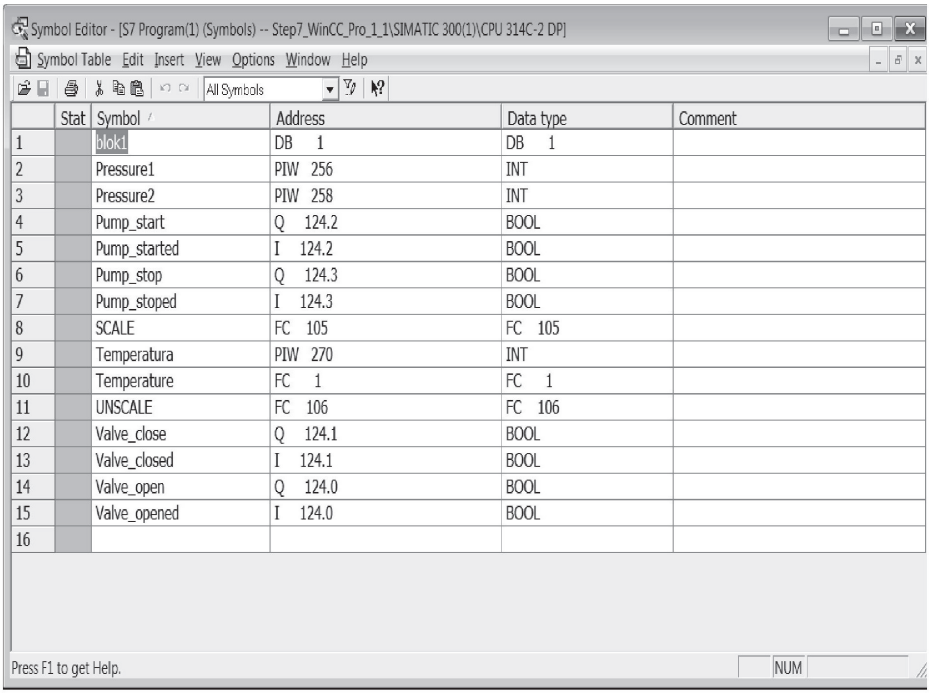


Рис. 2.1. Открытие таблицы символов

После этого можно добавлять в нее нужные адреса и присваивать им символьные имена (*рис. 2.2*).



	Stat	Symbol	Address	Data type	Comment
1		blok1	DB 1	DB 1	
2		Pressure1	PIW 256	INT	
3		Pressure2	PIW 258	INT	
4		Pump_start	Q 124.2	BOOL	
5		Pump_started	I 124.2	BOOL	
6		Pump_stop	Q 124.3	BOOL	
7		Pump_stoped	I 124.3	BOOL	
8		SCALE	FC 105	FC 105	
9		Temperatura	PIW 270	INT	
10		Temperature	FC 1	FC 1	
11		UNSCALE	FC 106	FC 106	
12		Valve_close	Q 124.1	BOOL	
13		Valve_closed	I 124.1	BOOL	
14		Valve_open	Q 124.0	BOOL	
15		Valve_opened	I 124.0	BOOL	
16					

Рис. 2.2. Таблица символов

2.3. Элементы программы

В среде Step7 будем работать со следующими элементами:

1) организационные блоки — OB;
2) функции — FC;
3) функциональные блоки — FB;
4) блоки данных — DB.

Каждому элементу присваивается порядковый номер, например OB1, OB2, FB1, FB2.

Назначение элементов следующее:

1) OB — организует взаимодействие программы с внешними выходами и входами;

2) FC — отвечает за выполнение типовых логических или математических операций;

3) FB — отвечает за логику работы отдельных типовых элементов системы (датчиков, механизмов и т. д.); при наличии в проекте мно-

жества элементов с одинаковой логикой работы можно разработать один раз FB, а затем тиражировать, используя разные DB, для хранения в них данных, полученных при каждом использовании FB;

4) DB — предназначен для хранения глобальных данных или данных, обработанных в функциональном блоке FB. Кроме того, каждой переменной, имеющейся в блоке данных, присваивается начальное значение, которое она будет иметь при загрузке блока в ЦПУ.

Порядок работы с элементами следующий:

1) описывается логика работы в FB или в FC;
2) создается DB, привязывается к FB;
3) в OB вставляется FB или FC и привязывается к нужным входам-выходам контроллера (через таблицу символов). Причем при каждом использовании FB нужно указывать новый DB, в котором будут храниться данные, полученные при обработке этого FB.

2.4. Работа с функцией FC

На примере аналоговых сигналов разберем работу функции FC.

Модуль аналогового ввода преобразует физический аналоговый сигнал (напряжение, ток, сопротивление, термо-ЭДС) в цифровой сигнал, пригодный для работы в ЦПУ. Цифровая переменная представляет собой число в формате Integer. Диапазон изменения этой цифровой переменной указан в руководстве по работе с сигнальными модулями. Пример данного диапазона для термопары типа L представлен в *табл.*

Данная переменная не может быть напрямую использована в операциях сравнения или математических операциях, поскольку она не несет в себе физического смысла. Поэтому нам нужно ее преобразовать в понятные нам градусы Цельсия.

Таблица

Представление аналоговых величин для термопар типа L

Тип L в °C	Единицы		Тип L в K	Единицы		Диапазон
	десятичные	16-ричные		десятичные	16-ричные	
>1150,0	32767	$7FFF_н$	> 1423,2	32767	$7FFF_н$	Переполнение

Окончание табл.

Тип L в °C	Единицы		Тип L в K	Единицы		Диапазон
	десятичные	16-ричные		десятичные	16-ричные	
1150,0 . . 901,0	11500 . . 9010	$2CEC_{H}$ . . 2332_{H}	1423,2 . . 1174,2	14232 . . 11742	3798_{H} . . $2DDE_{H}$	Перезагрузка
900,0 . —200,0	9000 . —2000	2338_{H} . $F830_{H}$	1173,2 . 73,2	11732 . 73,2	$2DD4_{H}$. $02DC_{H}$	Номинальный диапазон
< –200	< –2000	$<F830_{H}$	< 73,2	< 732	$< 02DC_{H}$	Отрицательное переполнение

Для ее преобразования выполним следующие операции.

1. Заходим в папку *Blocks* (*рис. 2.3*).

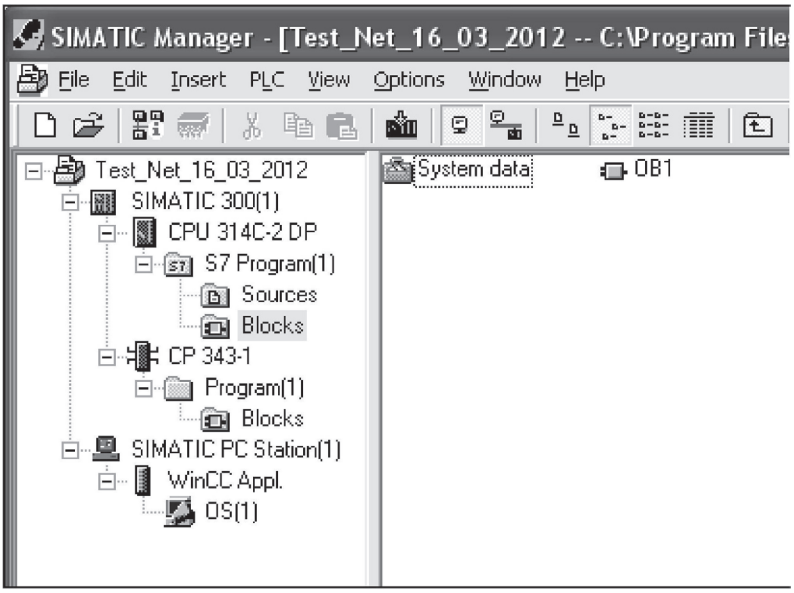


Рис. 2.3. Содержание папки Blocks

2. Создаем функцию: кликаем правой клавишей по свободному пространству и выбираем *Insert new object* > *Function* (*рис. 2.4*).

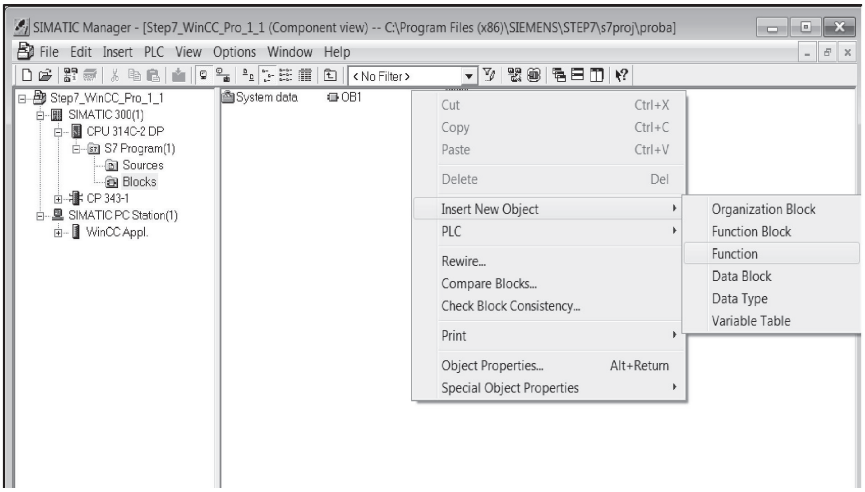


Рис. 2.4. Создание функции FC

3. Прописываем свойства функции в появившемся меню: присваиваем название, выбираем язык программирования (например, FBD), жмем *OK* (*рис. 2.5*).

4. Дважды кликаем левой клавишей мыши по появившейся функции FC1, на экране должны появиться следующие элементы, представленные на *рис. 2.6.*

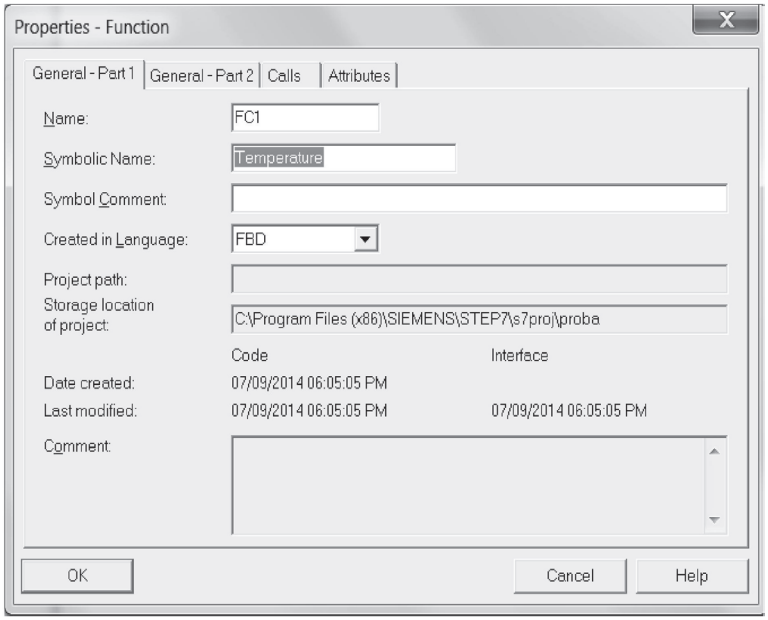


Рис. 2.5. Окно свойств функции FC

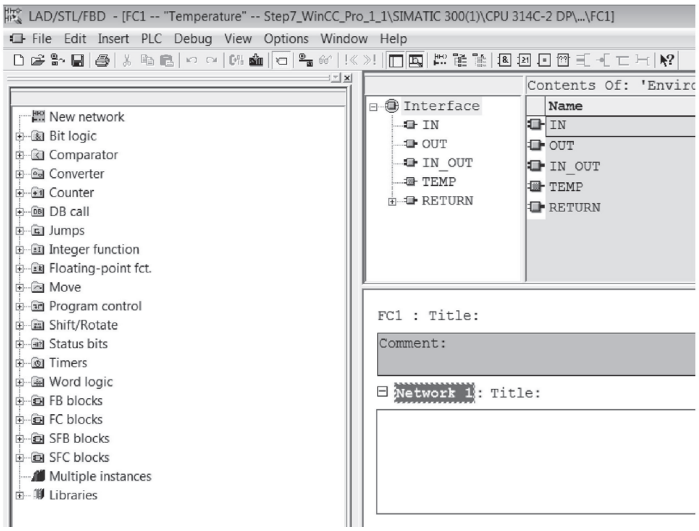


Рис. 2.6. Редактор блоков LAD/STL/FBD

5. Прописываем в папке *Interface* входные и выходные сигналы, другие локальные переменные для данного блока, присваиваем тип переменных (*рис. 2.7, 2.8*).

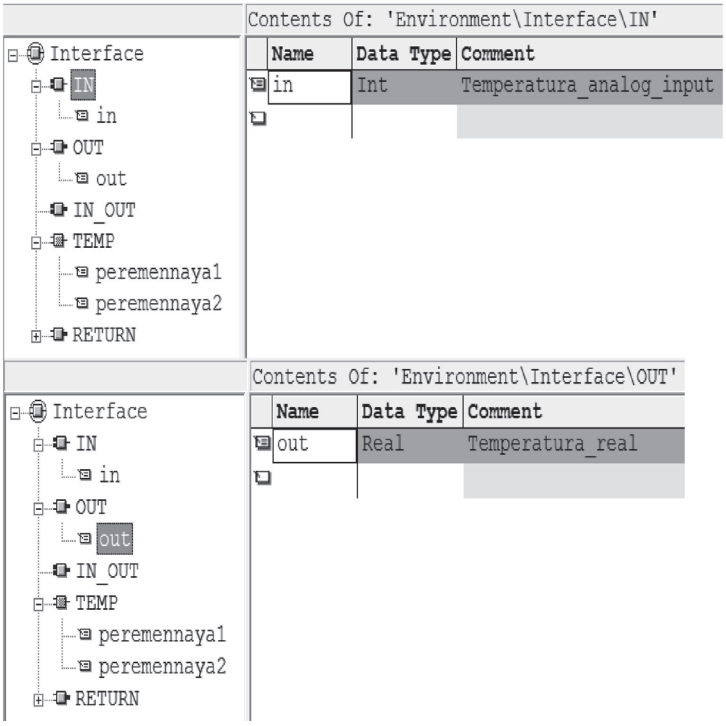


Рис. 2.7. Создание входных и выходных переменных функции

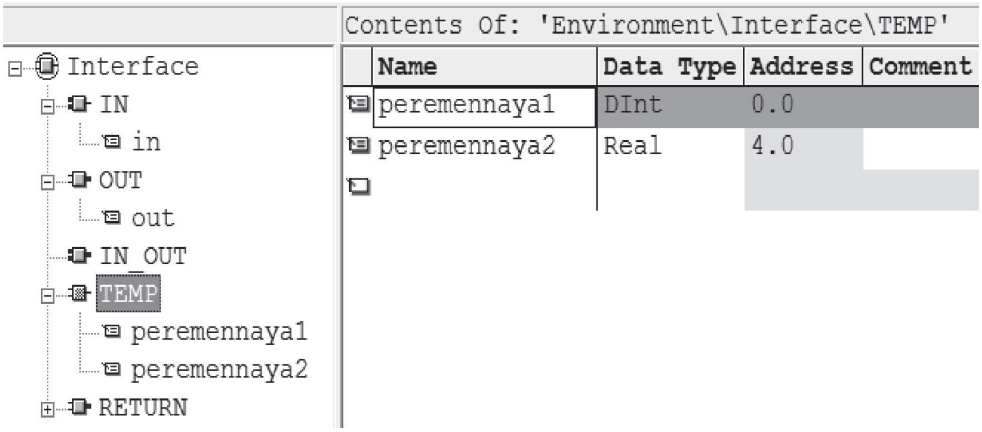


Рис. 2.8. Создание временных переменных функции

6. Из библиотеки выбираем нужные элементы для алгоритма работы функции и перетаскиваем их на рабочее поле.

7. Соединяем элементы между собой с помощью временных переменных, которые мы создали ранее. Присваиваем значения входам и выходам созданного алгоритма, для этого кликаем левой клавишей мыши по соответствующему элементу (обозначенному ??.?), начинаем вводить имя переменной и выбираем из выпадающего меню нужное имя (*рис. 2.9*).

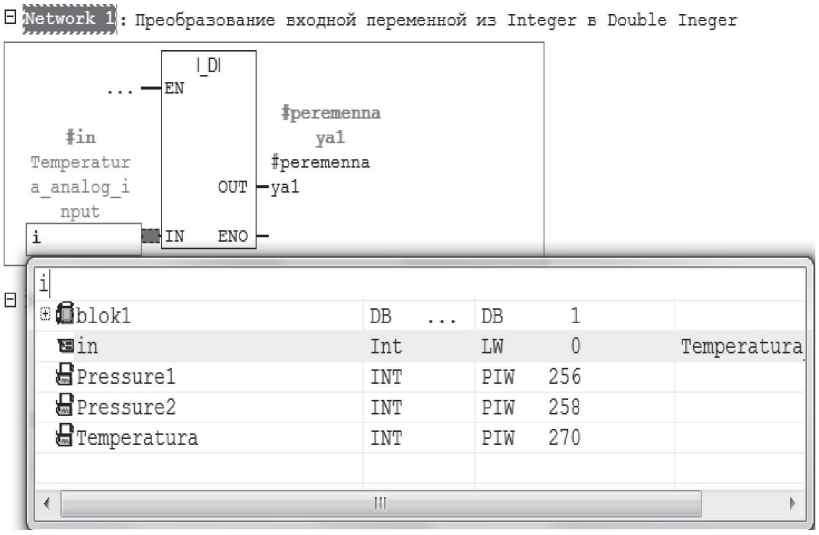


Рис. 2.9. Присвоение значений входам и выходам функции

8. Обратите внимание на то, что каждый блок необходимо поместить в новую сеть, для этого надо щелкнуть левой клавишей мыши по элементу Network в библиотеке. Необходимо строго соблюдать последовательность вычислений. Сначала мы преобразуем входную переменную из Integer в Double Integer, затем в Real и только потом производим преобразование в значение температуры. Как было видно из *табл.* на странице 42, для того чтобы получить значение температуры из той величины, которую выдал модуль аналогового ввода, необходимо эту величину разделить на 10 (*рис. 2.10*).

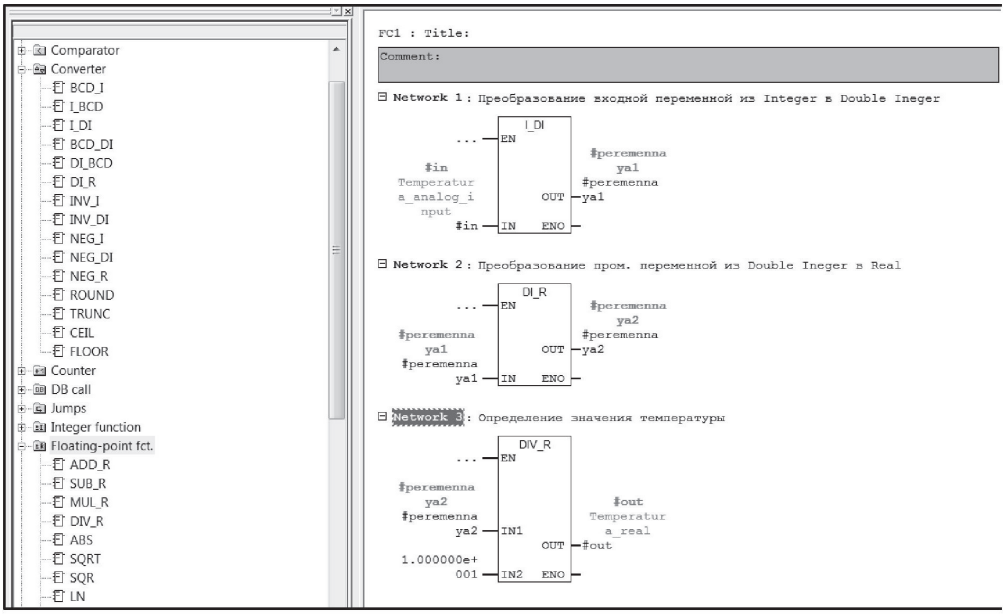


Рис. 2.10. Создание алгоритма функции

9. Закончив работу над алгоритмом, сохраняем его.

2.5. Работа с организационным блоком OB

Организационный блок OB1 обрабатывается процессором каждый цикл. Из данного организационного блока мы можем осуществить вызов созданной нами функции FC1. Для этого выполним следующие действия.

1. Заходим в OB1 (он, как правило, уже создан).
2. Выбираем в библиотеке в разделе FC blocks нужный нам FC.
3. Удерживая левой клавишей мыши, переносим его на рабочее поле (рис. *2.11*).

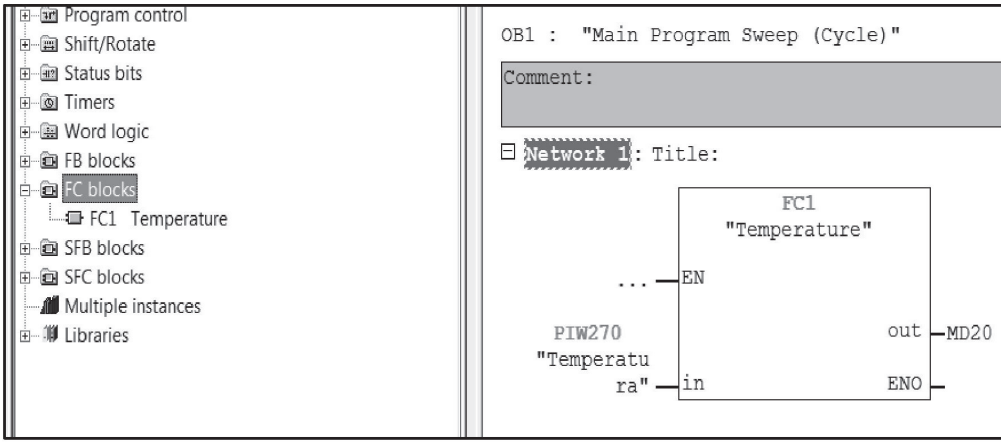


Рис. 2.11. Организационный блок OB1

4. Назначаем входы и выходы; обратите внимание на то, что на экране одновременно отображаются два значения адреса: имя символа (технологическое наименование сигнала) и соответствующий ему адрес модуля (входа или выхода). На выход функции подключим ячейку меркерной области памяти размером двойного слова MD20, поскольку на выходе данной функции мы имеем переменную типа *Real*, размер которой 32 бита.
4. Далее сохраняем и закрываем блок.

2.6. Считывание значения температуры с модуля аналогового ввода

Перед подключением термопары к модулю аналогового ввода его необходимо сконфигурировать. Отрываем HW Config, кликаем два раза на модуле аналогового ввода (*см. рис. 2.12*). Открывается окно свойств данного модуля. Открываем вкладку *Inputs*.

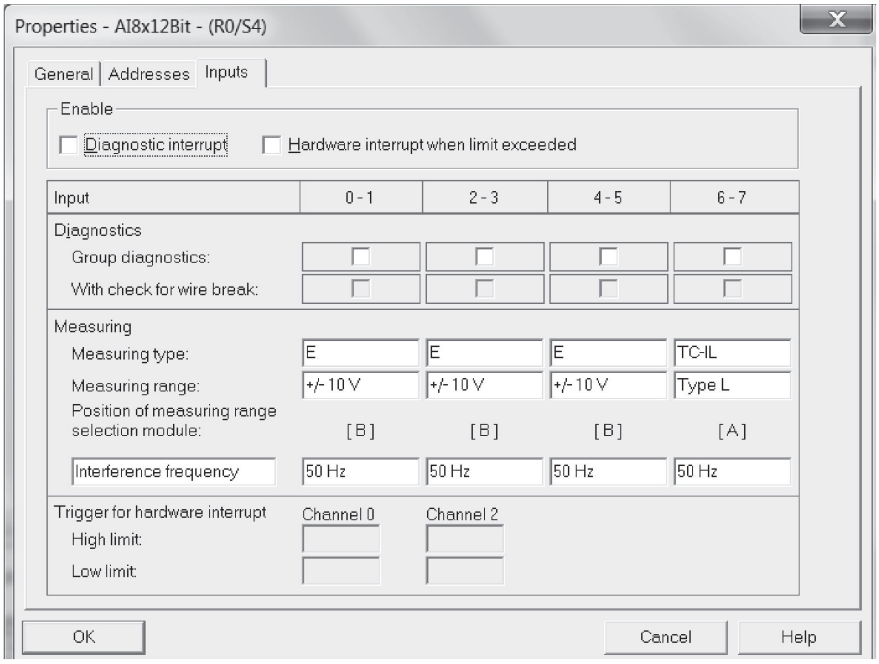


Рис. 2.12. Конфигурирование каналов модуля аналогового ввода

Для 6 и 7 каналов выберем в поле *Measuring type* (тип измерения) значение *TC–IL* — термоэлектрический преобразователь — внутренняя компенсация, линеаризация. Будем использовать внутреннюю компенсацию холодного спая. В поле *Measuring range* (диапазон измерения) выберем *Type L*, что соответствует подключаемой нами термопаре ТХК Метран-202.

Внутренняя компенсация холодного спая будет точной в том случае, если температура в помещении, где расположен холодный спай, будет такой же, как температура контроллера, или в том случае, если компенсационные провода будут подключаться напрямую к клеммам модуля аналогового ввода. В противном случае появится погрешность в измерении.

Затем необходимо скомпилировать и сохранить созданные изменения и загрузить их в контроллер. Как мы уже знаем, при загрузке конфигурации в контроллер он перейдет в *Stop*, после чего появится окно с предложением перевести ЦПУ в *RUN*.

После этого можно подключать термопару к модулю аналогового ввода. Подключение необходимо производить строго в соответствии с руководством по эксплуатации аналоговых модулей.

Поскольку для термопары мы сконфигурировали 6 и 7 каналы, для примера будем использовать канал 7. Подключаем концы от термопары на 18 и 19 клеммы модуля аналогового ввода, при этом необходимо соблюдать полярность.

Если в HW Config выделить модуль аналогового ввода и запустить в меню *PLC* команду *Monitor/Modify*, то в открывшемся окне можно увидеть, как изменяется значение аналогового входа 7, на который мы подключили термопару (*рис. 2.13*). Для этого нужно поставить галочку в поле *Monitor* и нагреть термопару.

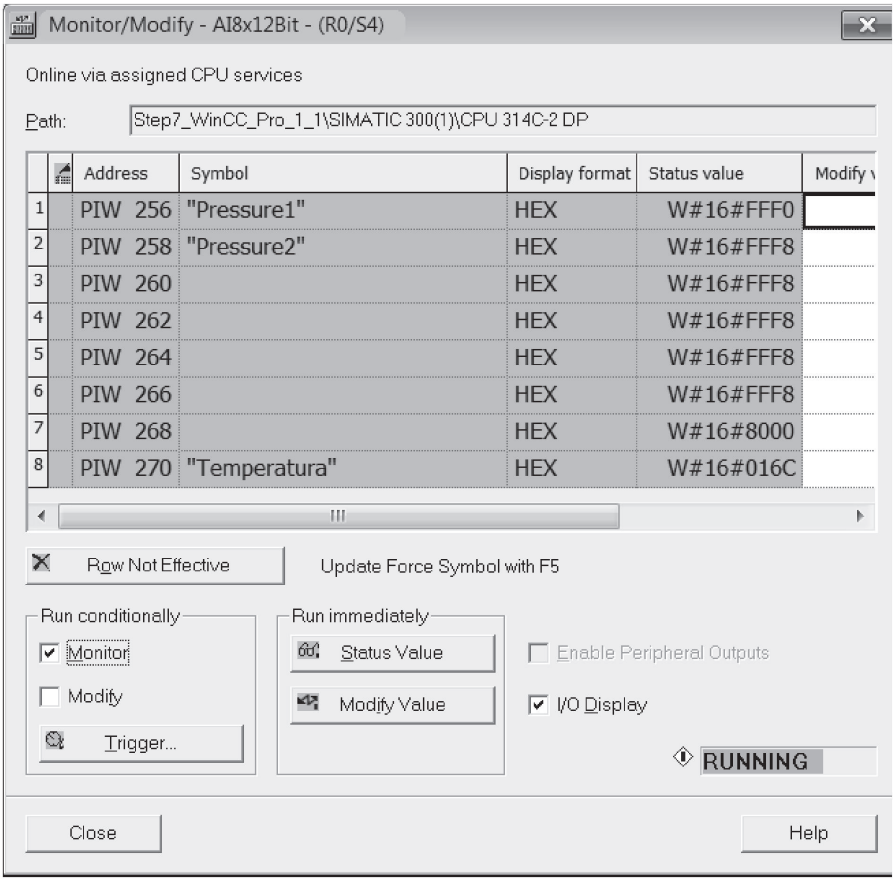


Рис. 2.13. Мониторинг изменения величины
на входах модуля аналогового ввода

Теперь загрузим в контроллер созданные нами организационный блок OB1 и функцию FC1. Необходимо помнить о том, что если мы сначала загрузим в контроллер блок OB1, а затем функцию FC1,

то ЦПУ перейдет в *Stop*. Это произойдет потому, что он начнет выполнять алгоритм блока OB1 и будет пытаться вызвать функцию FC1, но поскольку она не будет загружена в контроллер, то это будет считаться ошибкой, и ЦПУ перейдет в *Stop*. Поэтому нужно либо сначала загружать блоки, на которые имеются ссылки в других блоках, либо выделять все блоки, которые необходимо загрузить, и Simatic Manager сам определит, в какой последовательности их загружать в ЦПУ.

Если ЦПУ в режиме *RUN*, то мы можем посмотреть значение температуры, которое нам измеряет термопара. Для этого необходимо открыть блок OB1 и в меню *Debug* запустить команду *Monitor*. На выходе нашей функции появится значение температуры в градусах (*рис. 2.14*).

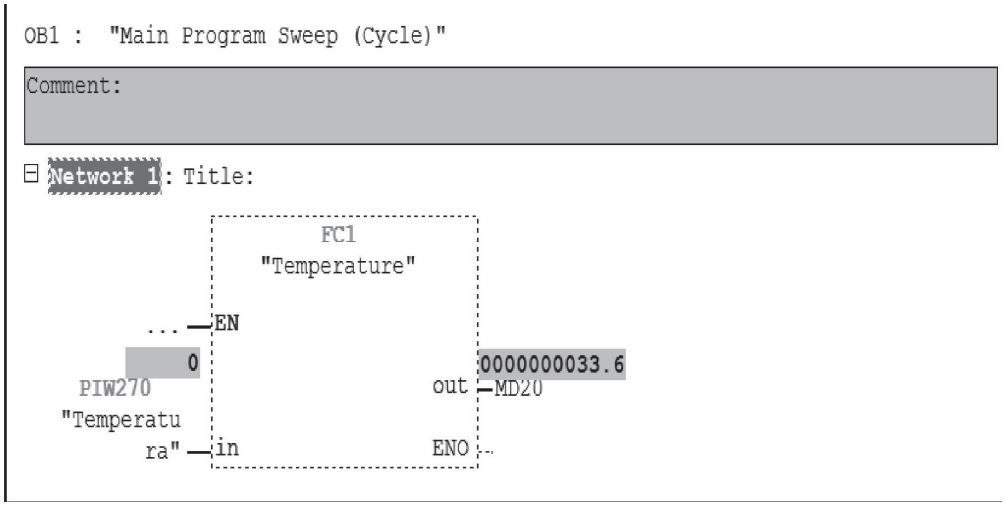


Рис. 2.14. Отображение значения температуры на выходе созданной нами функции FC1

2.7. Работа с функциональным блоком FB

На примере дискретных сигналов разберем работу функционального блока FB. Создадим функциональный блок FB1 для работы с задвижкой (*см. рис. 2.15*).

1. Заходим в папку *Blocks*.
2. Создаем функциональный блок: кликаем правой клавишей по свободному пространству, и выбираем *Insert new object > Function Block*.
3. Прописываем свойства блока в появившемся меню: присваиваем название, выбираем язык программирования (например, FBD), жмем *OK*.

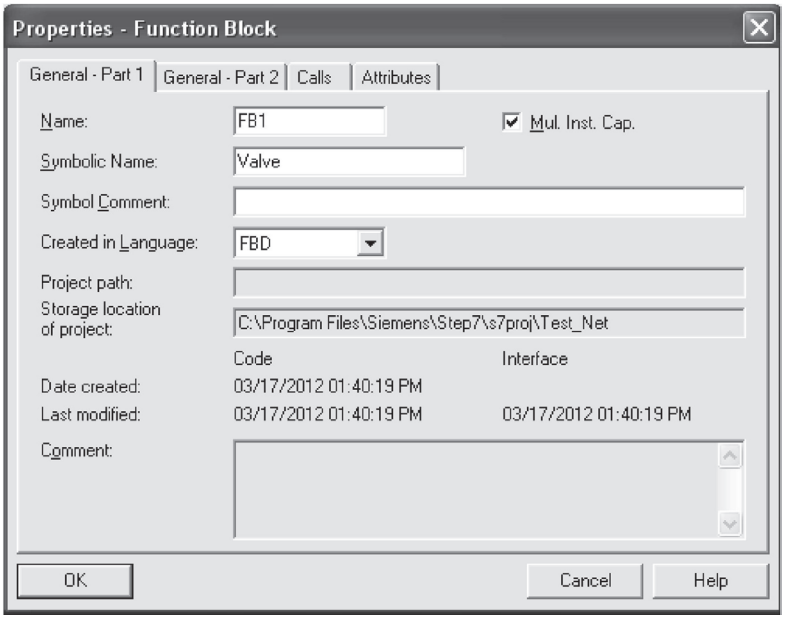


Рис. 2.15. Окно свойств функционального блока FB

4. Дважды кликаем левой клавишей мыши по появившемуся блоку FB1, на экране должна появиться такая картина (*рис. 2.16*).

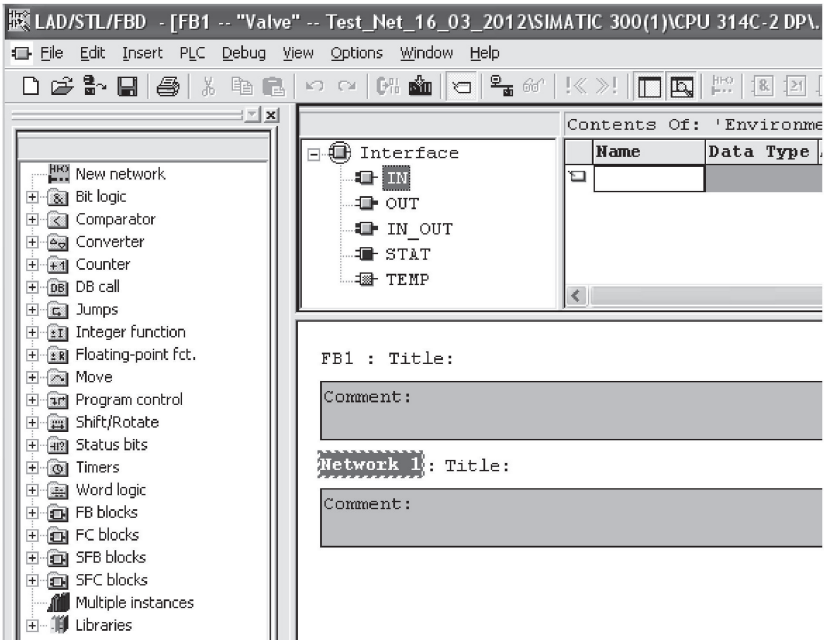


Рис. 2.16. Редактор блоков LAD/STL/FBD

Как и для функции FC, прописываем в папке *Interface* входные и выходные сигналы, другие локальные переменные для данного блока, присваиваем тип переменных.

5. Из библиотеки выбираем нужные элементы для алгоритма работы функционального блока и перетаскиваем их на рабочее поле.

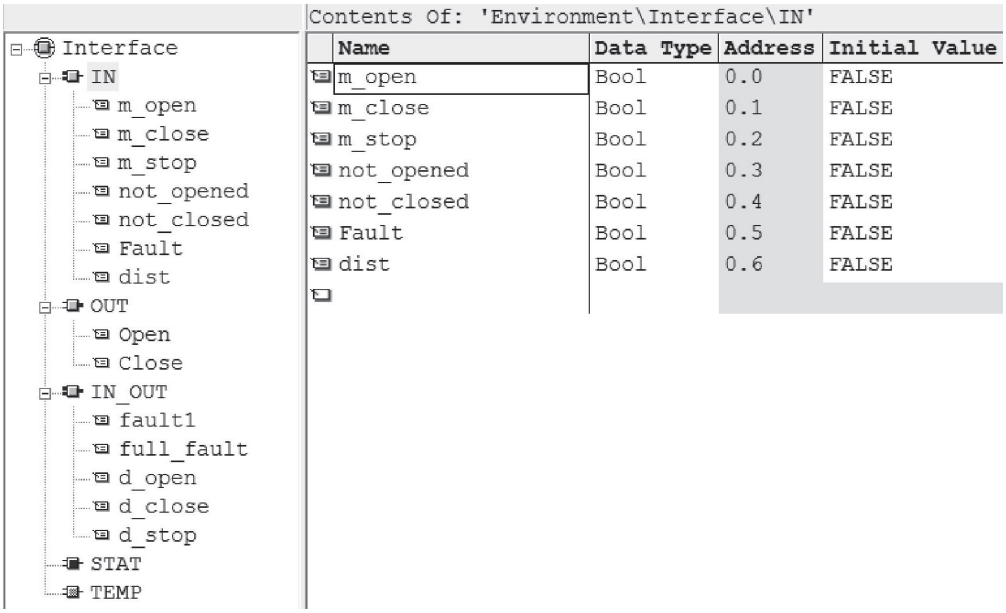


Рис. 2.17. Создание локальных переменных функционального блока

6. Соединяем элементы между собой с помощью временных переменных, которые мы создали ранее. Присваиваем значения входам и выходам созданного алгоритма, для этого кликаем левой клавишей мыши по соответствующему элементу (обозначенному ??.?), начинаем вводить имя переменной и выбираем из выпадающего меню нужное имя.

7. Обратите внимание на то, что каждую отдельную цепочку необходимо поместить в новую сеть, для этого надо щелкнуть левой клавишей мыши по элементу *Network* в библиотеке (*см. рис. 2.18*).

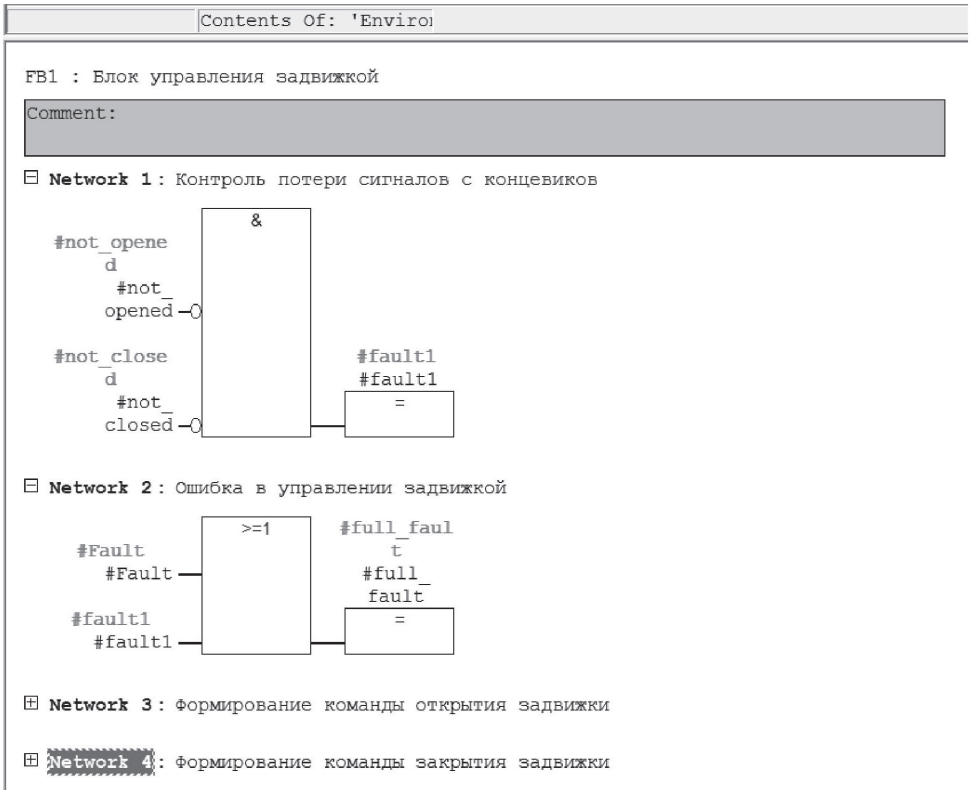


Рис. 2.18. Формирование сигнала о неисправности задвижки

Сигнал общей неисправности задвижки формируется в том случае, если одновременно отсутствуют сигналы с концевых выключателей *Не_открыто* и *Не_закрыто* или с задвижки приходит сигнал о ее неисправности (*рис. 2.19*).

Сигнал на открытие задвижки формируется, если имеется команда *Открытие по месту* и не выбран режим *Дистанционное управление* (т. е. включен режим Местное управление) или имеется команда *Открытие из ПТК* и выбран режим *Дистанционное управление*. Кроме того, должен отсутствовать сигнал на закрытие задвижки. Задвижка также остановится, если сработает концевой выключатель *Открыто*, или сформируется сигнал общей неисправности задвижки, или появится команда *Останов по месту* при выключенном режиме *Дистанционное управление*, или появится команда *Останов из ПТК* при включенном режиме *Дистанционное управление*.

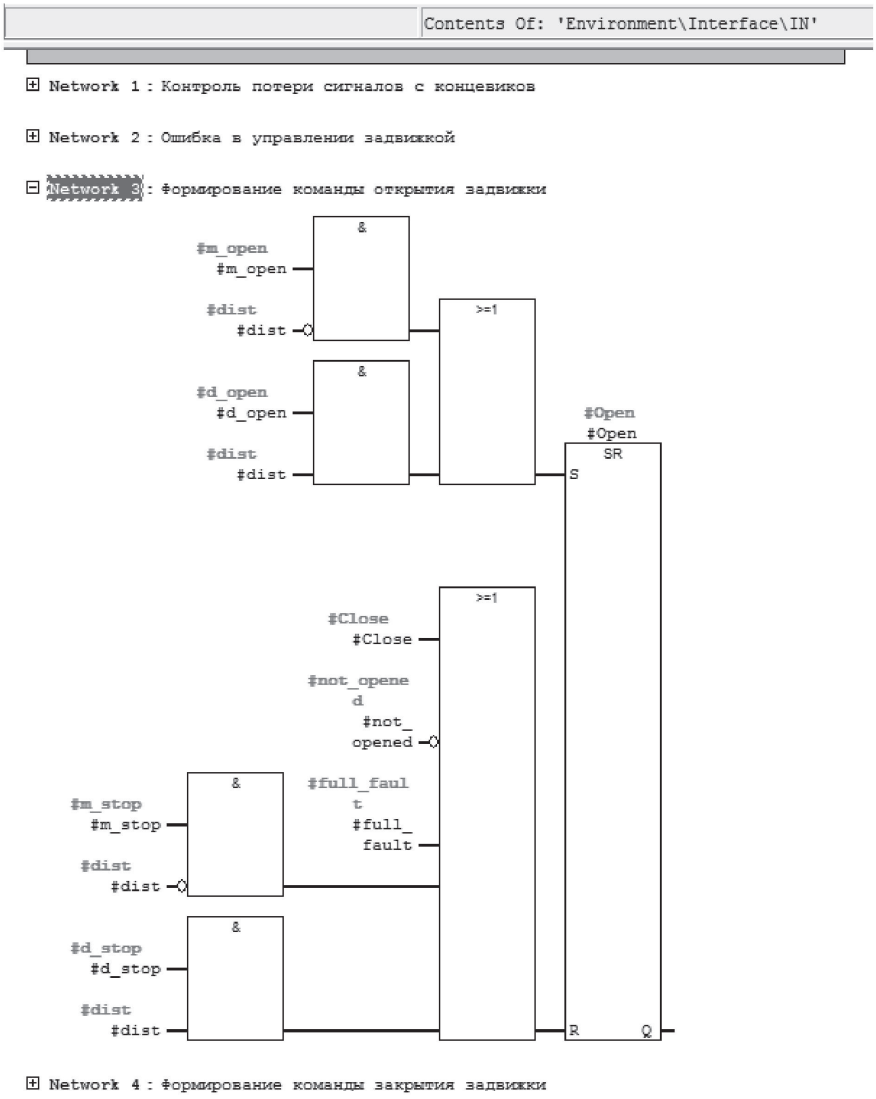


Рис. 2.19. Формирование команды открытия задвижки

Сигнал на закрытие задвижки формируется, если имеется команда *Закрытие по месту* и не выбран режим *Дистанционное управление* (т. е. включен режим *Местное управление*), или имеется команда *Закрытие из ПТК* и выбран режим *Дистанционное управление.* Кроме того, должен отсутствовать сигнал на открытие задвижки. Задвижка также остановится, если сработает концевой выключатель *Закрыто*, или сформируется сигнал общей неисправности задвижки, или появится команда

Останов по месту при выключенном режиме *Дистанционное управление*, или появится команда *Останов из ПТК* при включенном режиме *Дистанционное управление* (*рис. 2.20*).

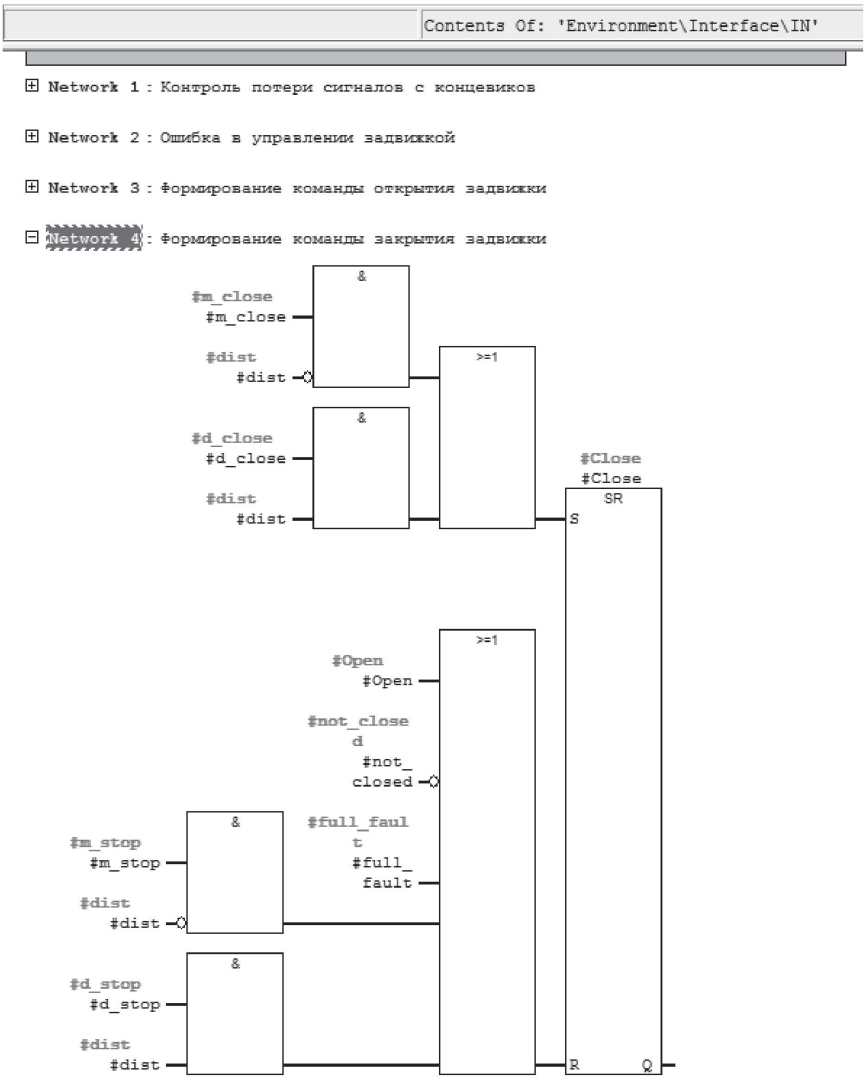


Рис. 2.20. Формирование команды закрытия задвижки

8. Закончив работу над алгоритмом, сохраняем данные.

9. Теперь необходимо организовать вызов функционального блока FB1 из организационного блока OB1. Для этого открываем OB1 и создаем новую сеть.

10. Вытаскиваем на рабочее поле созданный нами функциональный блок FB1 (*рис. 2.21*).

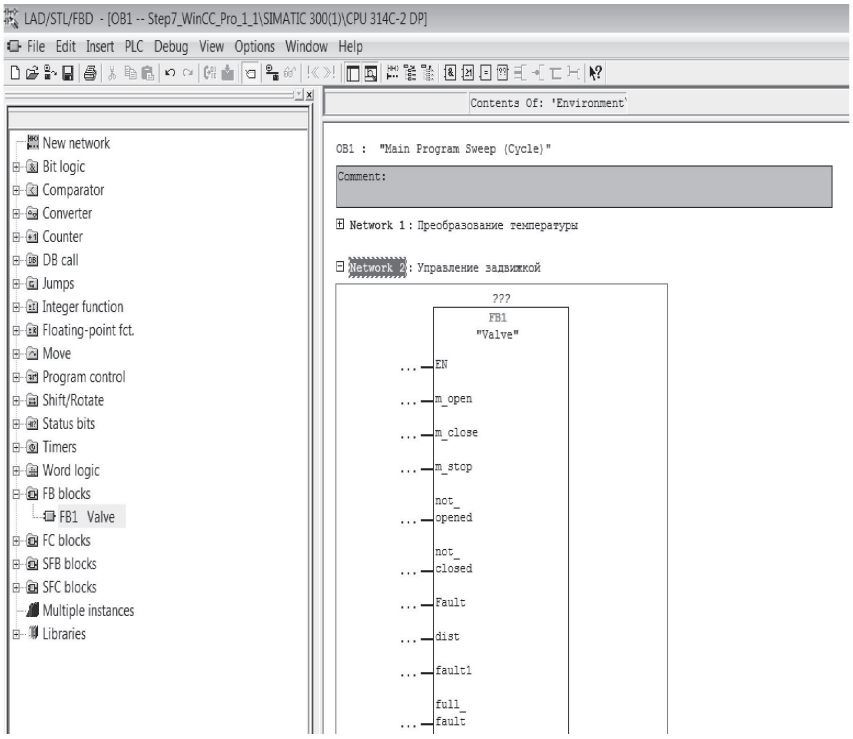


Рис. 2.21. Добавление блока FB1 на рабочее поле блока OB1

11. Вместо знаков ??.? задаем имя экземплярного блока данных DB1 и нажимаем Enter. После этого появляется окно с информацией о том, что данный экземплярный блок данных не создан и предлагается создать его. Нажимаем YES (*рис. 2.22*).

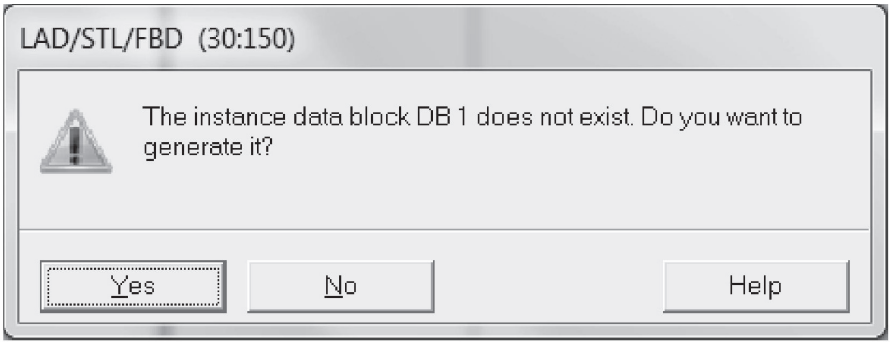
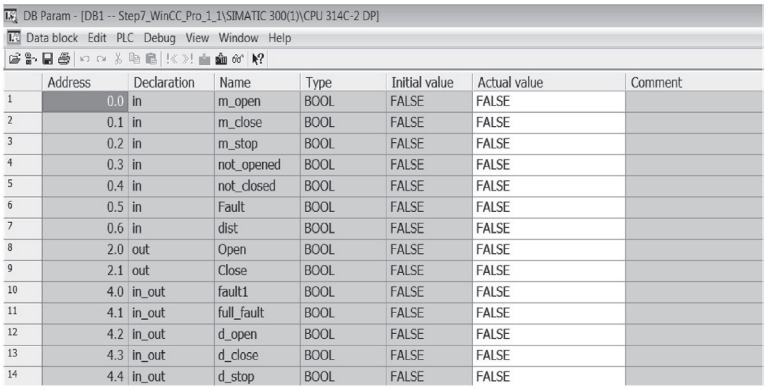


Рис. 2.22. Создание экземплярного блока данных

12. Теперь в папке *Blocks* появился блок данных DB1. Он содержит все входные (IN), выходные (OUT) и проходные (IN-OUT) переменные, которые были созданы в функциональном блоке FB1 (*рис. 2.23*).


DB Param - [DB1 -- Step7_WinCC_Pro_1_1\SIMATIC 300(1)\CPU 314C-2 DP]

Data block Edit PLC Debug View Window Help

	Address	Declaration	Name	Type	Initial value	Actual value	Comment
1	0.0	in	m_open	BOOL	FALSE	FALSE	
2	0.1	in	m_close	BOOL	FALSE	FALSE	
3	0.2	in	m_stop	BOOL	FALSE	FALSE	
4	0.3	in	not_opened	BOOL	FALSE	FALSE	
5	0.4	in	not_closed	BOOL	FALSE	FALSE	
6	0.5	in	Fault	BOOL	FALSE	FALSE	
7	0.6	in	dist	BOOL	FALSE	FALSE	
8	2.0	out	Open	BOOL	FALSE	FALSE	
9	2.1	out	Close	BOOL	FALSE	FALSE	
10	4.0	in_out	fault1	BOOL	FALSE	FALSE	
11	4.1	in_out	full_fault	BOOL	FALSE	FALSE	
12	4.2	in_out	d_open	BOOL	FALSE	FALSE	
13	4.3	in_out	d_close	BOOL	FALSE	FALSE	
14	4.4	in_out	d_stop	BOOL	FALSE	FALSE	

Рис. 2.23. Экземплярный блок данных

13. Подключаем нужные нам выходы и входы (*рис. 2.24*).

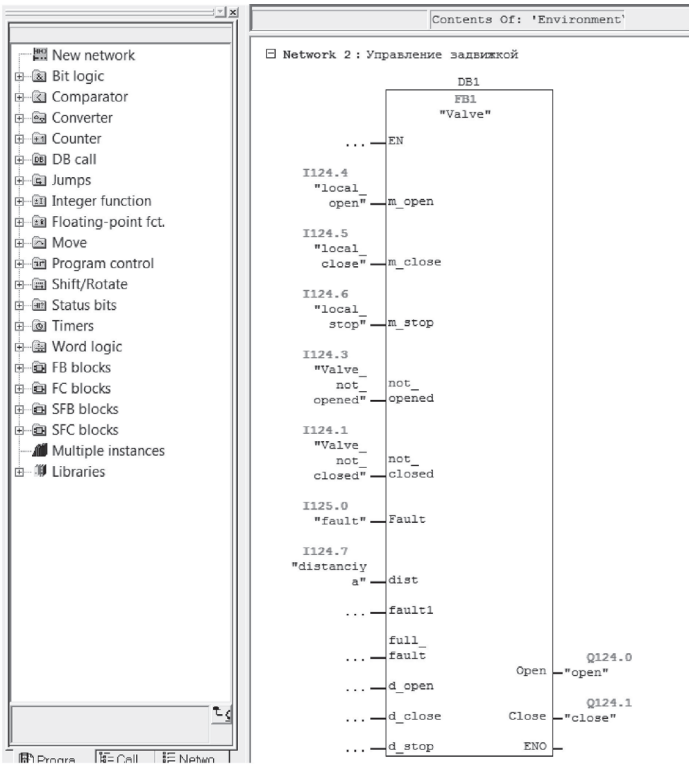


Рис. 2.24. Подключение входов и выходов к функциональному блоку FB1

14. Сохраняем OB1 и загружаем его в контроллер.

2.8. Подключение стенда к контроллеру и проверка работы блока FB1

Теперь нужно проверить работу созданного нами блока FB1, для этого подключим к контроллеру кнопки управления, концевые выключатели и контакты пускателя привода задвижки.

Подключение производим в соответствии с таблицей символов, которую мы создали ранее (*рис. 2.2*).

I 124.0 — Концевой выключатель *Закрыто*
I 124.1 — Концевой выключатель *Не_закрыто*
I 124.2 — Концевой выключатель *Открыто*
I 124.3 — Концевой выключатель *Не_открыто*
I 124.4 — Кнопка *Открыть* по месту
I 124.5 — Кнопка *Закрыть* по месту
I 124.6 — Кнопка *Стоп* по месту
I 124.7 — Переключатель *Дистанционное управление* по месту
I 125.0 — Контакт задвижки *Авария*
Q 124.0 — Клеммы пускателя *Открыть*
Q 124.1 — Клеммы пускателя *Закрыть*

Концевые выключатели и контакт *Авария* заменяем тумблерами, а клеммы пускателя — лампочками.

Подключение необходимо производить строго в соответствии с руководством по эксплуатации контроллера VIPA CPU 314SC.

Для дискретных входов +24 В источника питания PS 307 необходимо через соответствующий контакт кнопки или концевого выключателя подключить к соответствующему дискретному входу контроллера (клемма 2 для I124.0, 3 — I124.1, 9 — I124.7 и 12 — I125.0). Кроме того, необходимо +24 В подключить на клемму 1, а –24 В — на клемму 20 для питания схемы модуля дискретных входов.

Для дискретных выходов +24 В источника питания PS 307 необходимо подключить на 21 и 31 клемму, а –24 В — на клемму 30 и 40 для питания схемы модуля дискретных выходов. Кроме того, необходимо –24 В подключить к минусовым клеммам ламп, которые заменяют пускатель привода задвижки, а плюсовые клеммы ламп — к соот-

ветствующим контактам модуля дискретных выходов (клемма 22 для Q124.0 и 23 для Q124.1).

Далее необходимо проверить работу блока FB1. Для этого переведем переключатель *Дистанционное управление* в положение *Отключено* (включен режим *Местное управление*), тумблер *Авария* в положение *Отключено*, тумблер концевика *Открыто* в положение *Не_открыто* и тумблер концевика *Закрыто* в положение *Не_закрыто*.

Задвижка находится в промежуточном положении, нажимаем кнопку *Открыть*, должна загореться лампочка *Открытие*. Лампа будет гореть до тех пор, пока мы не нажмем кнопку *Стоп*, не переключим тумблер концевика *Открыто* в положение *Открыто* или не переключим тумблер *Авария* в положение *Включено*. Если мы выполним одно из вышеперечисленных действий, то лампа погаснет, т. е. задвижка остановится. Пока задвижка идет на открытие, команда на закрытие не срабатывает.

Для команды *Закрыть* действия по проверке аналогичны.

2.9. Работа с глобальным блоком данных DB

В глобальном блоке данных можно хранить результаты вычислений созданных нами функций, результаты преобразований входных аналоговых величин из формата *Integer* в формат *Real* и другие данные, полученные в результате каких-либо вычислений, которые необходимо сохранить.

Работу с глобальным блоком данных рассмотрим на примере аналоговых и дискретных сигналов. Создадим блок данных DB2 для хранения в нем информации, полученной с клапана и передаваемой на клапан.

1. Заходим в папку Blocks.
2. Создаем блок данных: кликаем правой клавишей по свободному пространству и выбираем *Insert new object > Data Block*.
3. Прописываем свойства блока в появившемся меню: присваиваем название, устанавливаем тип — *Shared DB*, жмем *OK* (*рис. 2.25*).

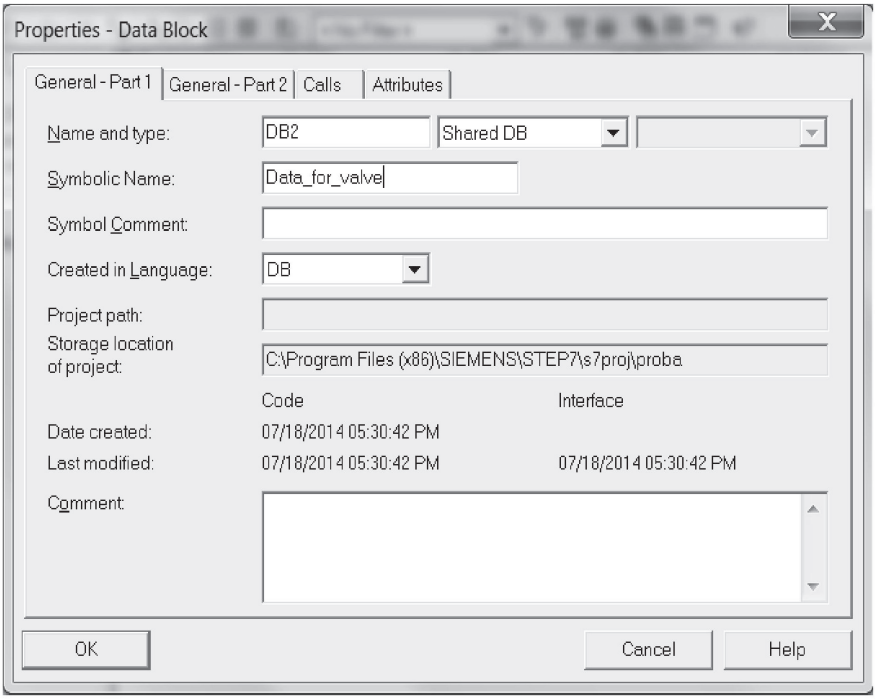


Рис. 2.25. Создание глобального блока данных

4. Открываем блок данных DB2 и создаем необходимые нам переменные (*рис. 2.26*).

DB2 -- "Data_for_valve" -- Step7_WinCC_Pro_1_1\SIMATIC 300(1)\CPU 314C-2 DP\...\DB2

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	HI_lim_UP	REAL	1.000000e+002	
+4.0	LOW_lim_UP	REAL	0.000000e+000	
+8.0	BIPOLAR_UP	BOOL	FALSE	
+10.0	Valve_UP	REAL	0.000000e+000	
+14.0	up_error_UP	WORD	W#16#0	
+16.0	Valve_out	REAL	5.000000e+001	
+20.0	up_error_OUT	WORD	W#16#0	
+22.0	HI_lim_OUT	REAL	1.000000e+002	
+26.0	LOW_lim_out	REAL	0.000000e+000	
+30.0	BIPOLAR_OUT	BOOL	FALSE	
+30.1	Valve_opened	BOOL	FALSE	
+30.2	Valve_closed	BOOL	FALSE	
=32.0		END_STRUCT		

Рис. 2.26. Создание переменных глобального блока данных

В данном блоке данных будут храниться:

- верхние и нижние пределы указателя положения клапана и управляющего сигнала клапана;
- биты выбора биполярности сигнала для указателя положения и управляющего сигнала;
- значения указателя положения и управляющего сигнала, полученные после преобразования в функциях FC105 и FC106;
- информация об ошибке преобразования сигналов указателя положения и управляющего сигнала с помощью функций FC105 и FC106, если она возникнет;
- состояние концевых выключателей клапана.

5. Сохраняем и закрываем данный блок данных.

6. Открываем организационный блок OB1 и создаем новый *Network*.

7. Из библиотеки выбираем функции FC105 и FC106 (*Libraries* > *Standard Library* > *TI-S7 Converting Blocks*), назначаем входы и выходы согласно *рис. 2.27, 2.28.*

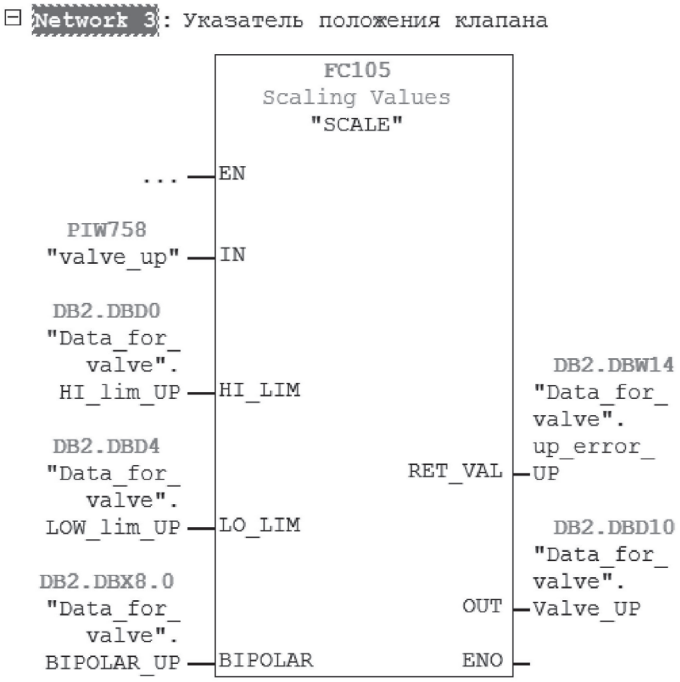


Рис. 2.27. Назначение входов и выходов функции FC105 для преобразования сигнала указателя положения

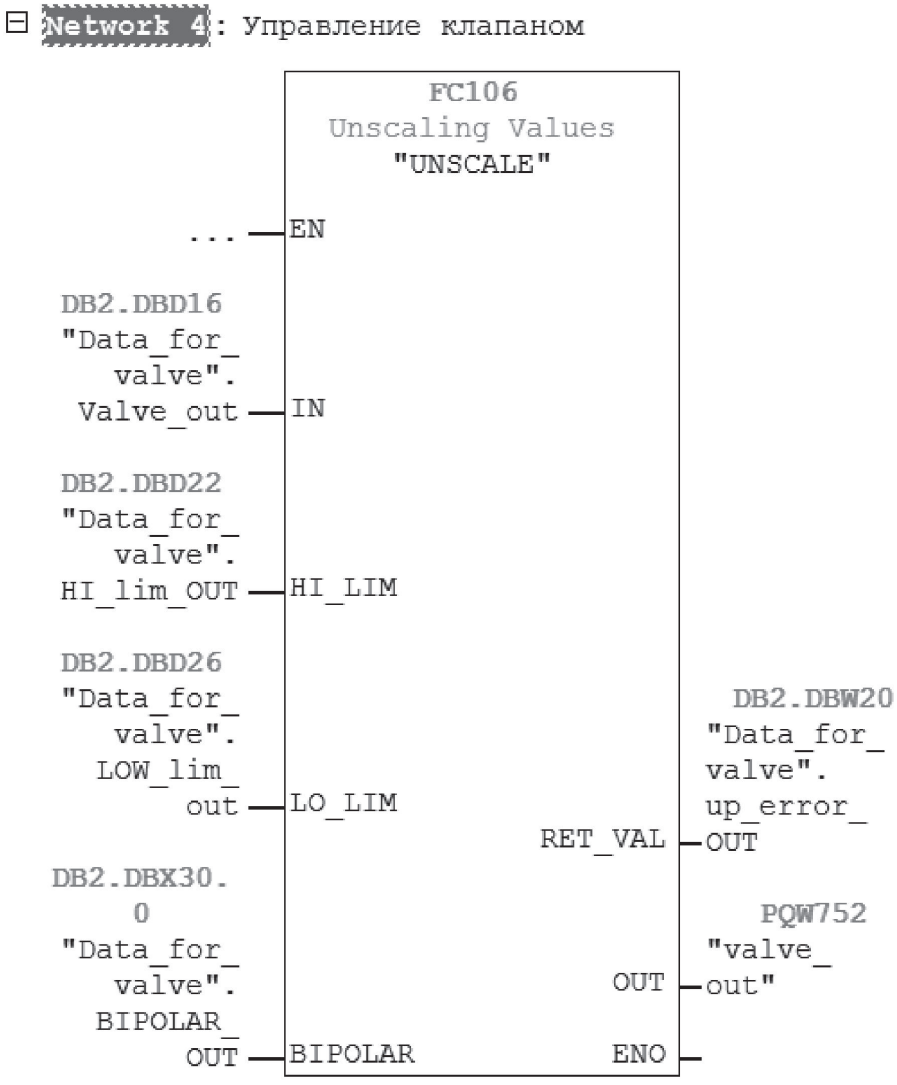


Рис. 2.28. Назначение входов и выходов функции FC106 для преобразования управляющего сигнала

8. Организуем запись значений концевых выключателей в DB2 (*рис. 2.29*).

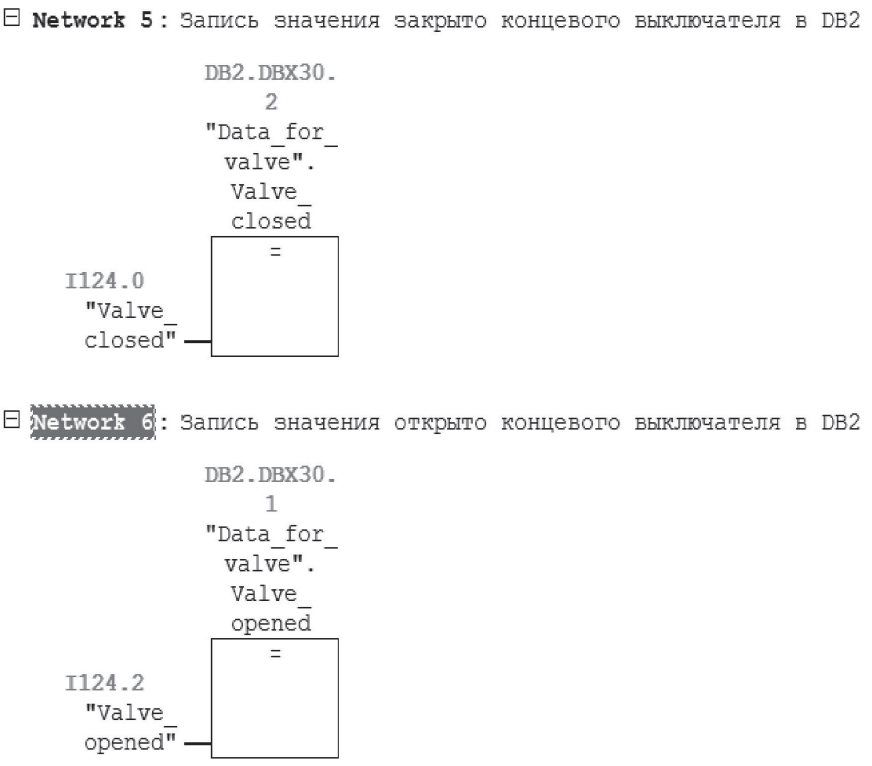


Рис. 2.29. Запись значений концевых выключателей в DB2

9. Сохраняем и закрываем блок OB1.

Перед загрузкой алгоритма в контроллер его нужно проверить в симуляторе.

2.10. Работа с симулятором PLCSIM

Созданную нами программу перед загрузкой в контроллер необходимо проверить на симуляторе, для того чтобы отследить и исправить ошибки, которые практически всегда возникают при создании программы, не допустив при этом нарушений в управлении технологическим оборудованием.

Для этого выбираем в меню *Options* функцию *Simulate module*.

1. Загружаем симулятор.

2. Заходим в *Options* и создаем на рабочем пространстве необходимое нам количество имитаторов входных и выходных модулей контроллера, используя функции *Input Variable* и *Output Variable*.

3. Присваиваем модулям адреса, соответствующие реальным (прописанным в Hardware), например: PIW 758, PQW 752, IB 124 (*рис. 2.30*).

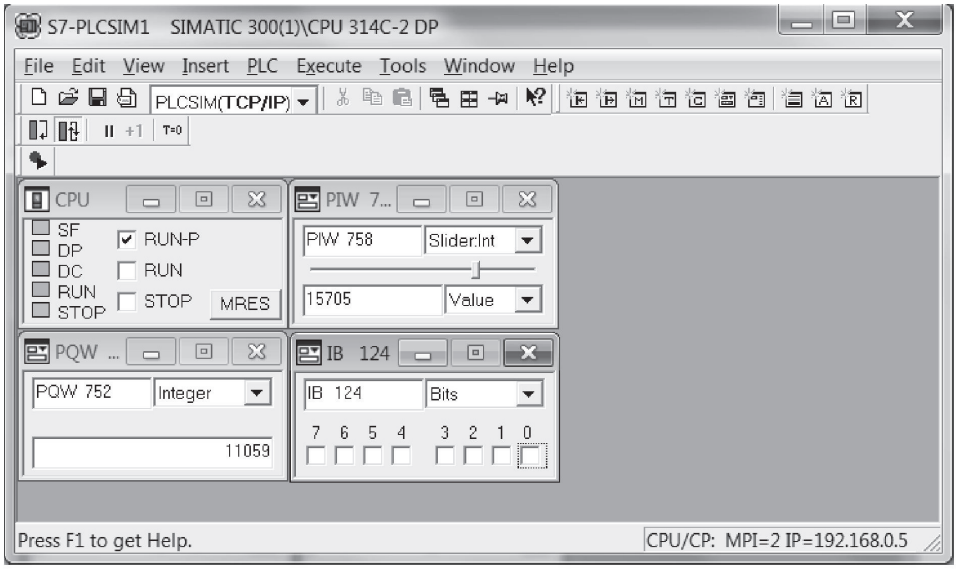


Рис. 2.30. Симулятор S7-PLCSIM

4. Присоединяем к модулям таблицу символов (*рис. 2.31*).

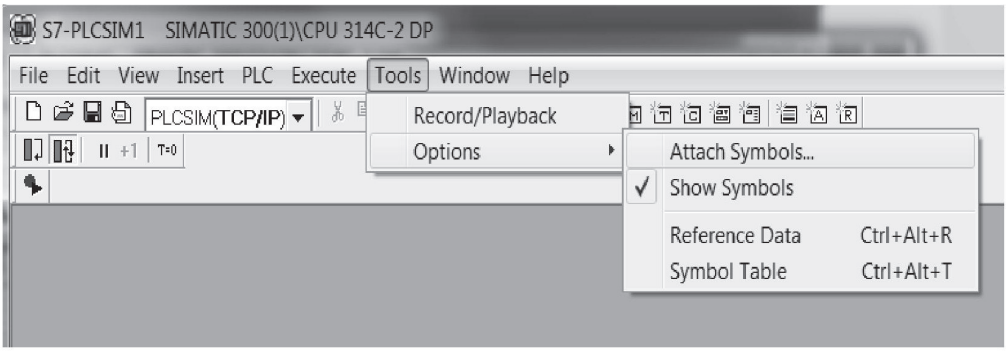


Рис. 2.31. Присоединение таблицы символов

Для этого находим ее в проводнике появившегося окна *Open* и жмем *OK* (*рис. 2.32*).

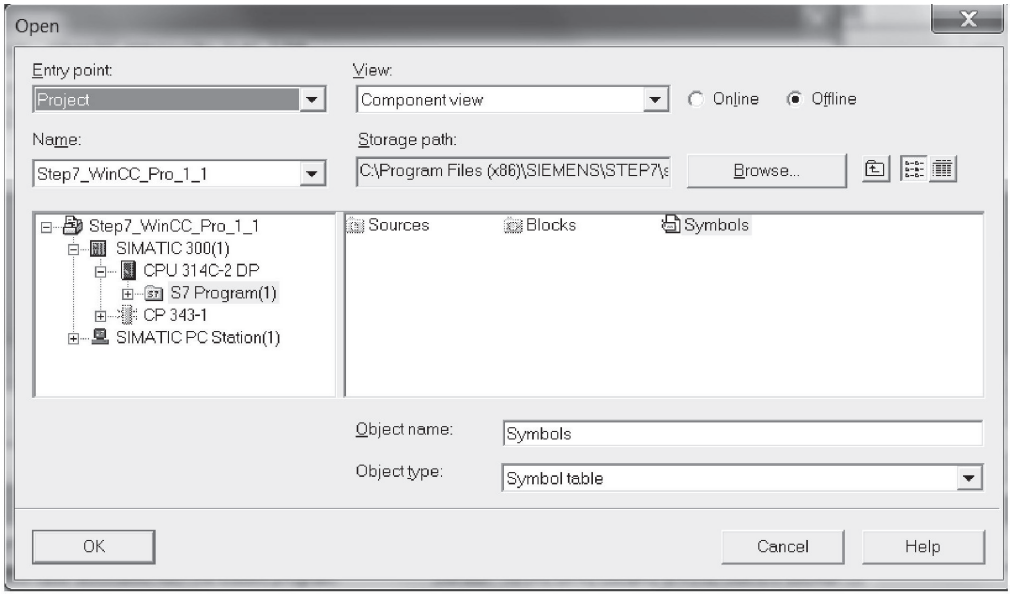


Рис. 2.32. Выбор нужной таблицы символов

5. Теперь при наведении курсором на адрес модуля будет всплывать подсказка с именем символа, что очень удобно при испытаниях программы.

6. Обратите внимание, что IB и QB — это имитаторы дискретных входов и выходов контроллера, а PIW и PWQ — имитаторы аналоговых входов и выходов контроллера.

7. Закончив набор модулей, загружаем написанную нами программу в симулятор, для этого возвращаемся к папке *Blocks* и нажимаем кнопку *Загрузить*.

8. Работу программы удобно отслеживать, когда на экране одновременно находятся симулятор и открытый блок, чью работу проверяем. При этом надо выполнить следующие действия: перевести симулятор в режим *Run*, а в блоке включить режим *Online*. Внизу экрана блока должен появиться мигающий индикатор зеленого цвета (*рис. 2.33*).

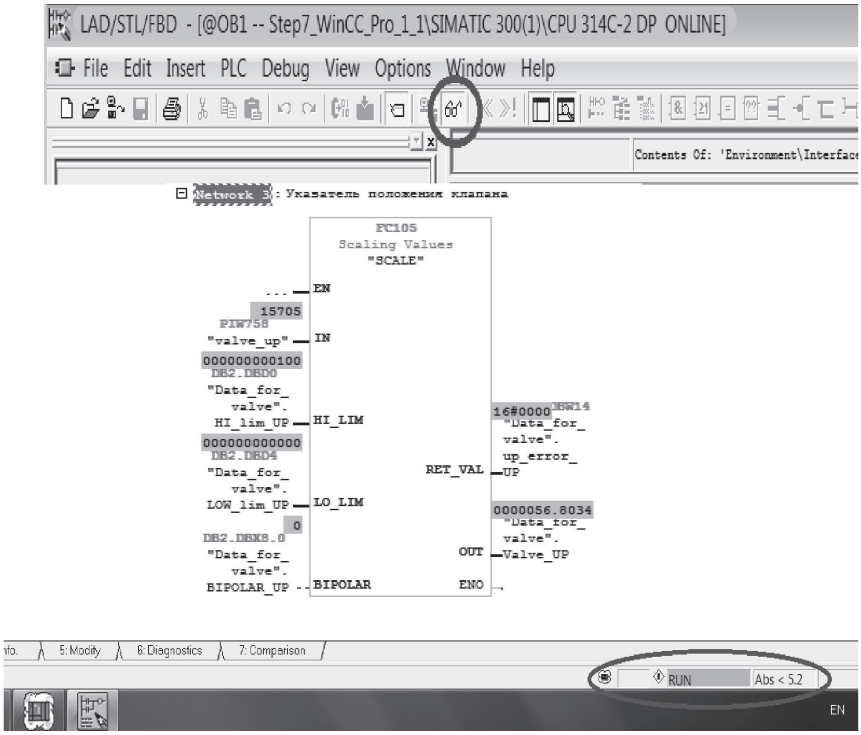


Рис. 2.33. Включение режима Online в Step7 и Run в S7-PLCSIM

9. Теперь мы можем отслеживать реакцию блока на имитацию входных сигналов и проверять, правильно ли собрана логика (*рис. 2.34*).

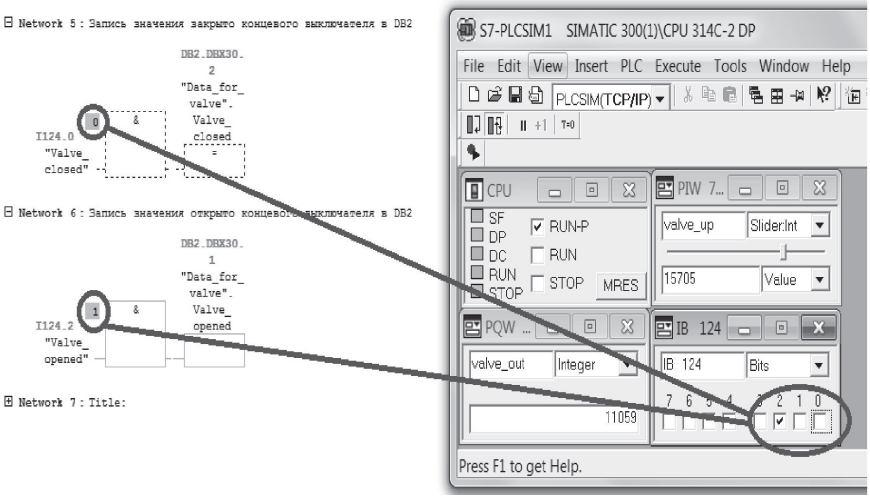


Рис. 2.34. Реакция блока OB1 на изменение значения дискретного входа в S7-PLCSIM

10. Если видим, что блок прописан неправильно, то можем внести изменения; для этого переходим в *Offline*, останавливаем контроллер, вносим необходимые изменения, загружаем блок снова (соглашаемся на все условия в выпадающих меню) и снова запускаем симуляцию.

После того как алгоритм проверен в симуляторе, можно его загружать в контроллер и подключать входы и выходы клапана к модулям ввода и вывода.

ЧАСТЬ 3. СОЗДАНИЕ ПРОЕКТА В WINCC

В этой главе мы будем рассматривать вопросы по организации совместной работы человеко-машинного интерфейса WinCC и исполнительной среды Step7. Вопросы по работе в самой программе WinCC (работа в графическом редакторе, запуск симулятора, построение отчетов и др.) достаточно хорошо изложены в учебном пособии Siemens AG «WinCC. Начало работы» (2003 г.), подробно на них останавливаться не будем.

После того как закончена работа в Step7, необходимо скомпилировать PS-станцию, чтобы все необходимые данные из Step7 были доступны в WinCC. Для этого нужно нажать правой кнопкой мыши на *SIMATIC PS Station* и выбрать команды *PLS/Compile and Download Objects* (*рис. 3.1*).

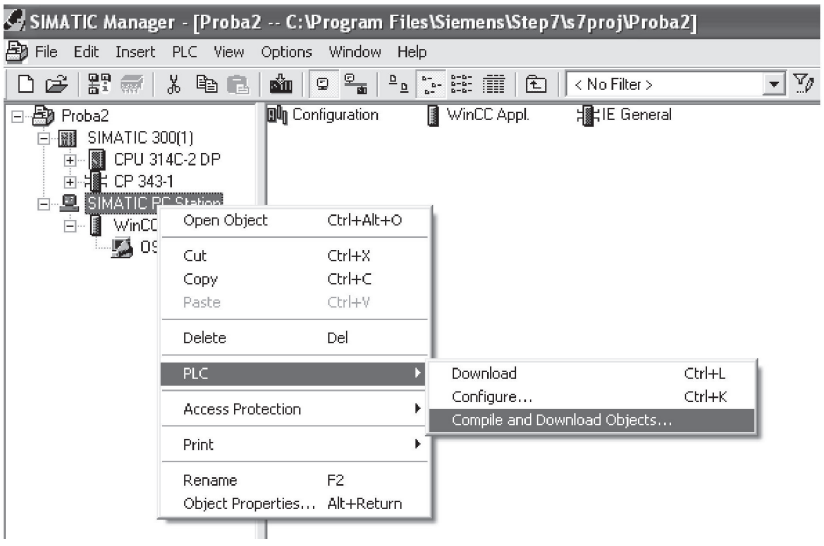


Рис. 3.1. Открытие окна компиляции PS-станции

После этого откроется окно компиляции, в котором нужно отметить галочками объекты, которые мы хотим скомпилировать (*рис. 3.2*).

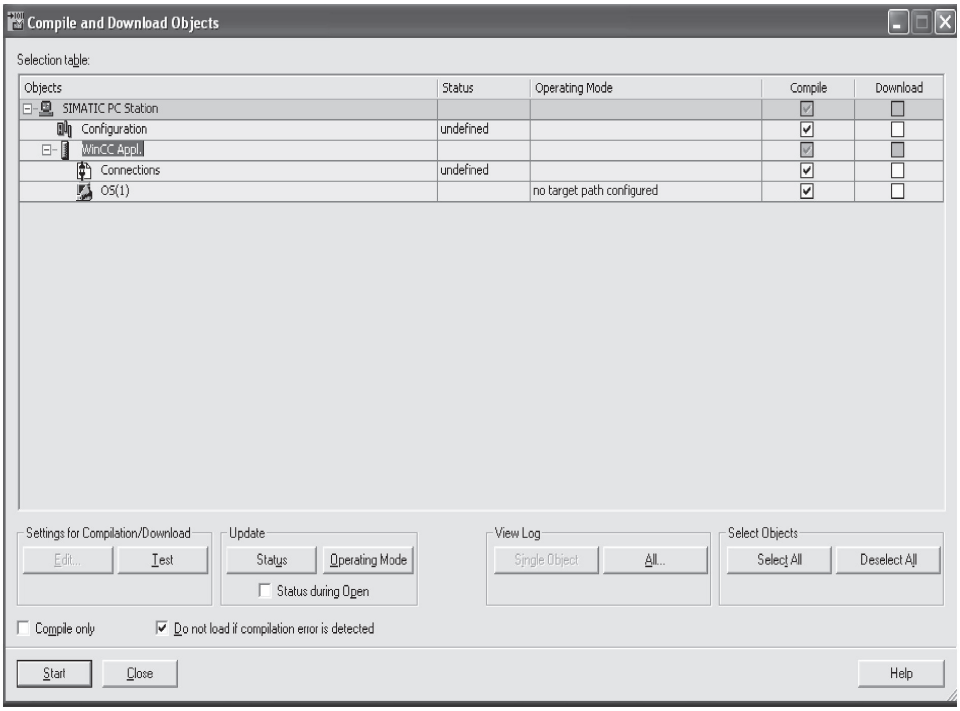


Рис. 3.2. Окно компиляции PS-станции

Поскольку в нашем случае OS-станция (станция оператора) находится на том же компьютере, что и ES-станция (инженерная станция), то проект загружать не требуется, так как он уже расположен на нашем компьютере.

Нажимаем на кнопку *Start*, затем в открывшемся окне нажимаем *Yes*, и процесс компиляции запускается.

В процессе компиляции появится запрос о выборе назначения уникальных номеров сообщений. Оставляем настройки неизменными и нажимаем *OK* (*рис. 3.3*).

После завершения компиляции откроется текстовый файл, в котором будет написано, что программа скомпилирована без ошибок.

После этого окно компиляции можно закрыть.

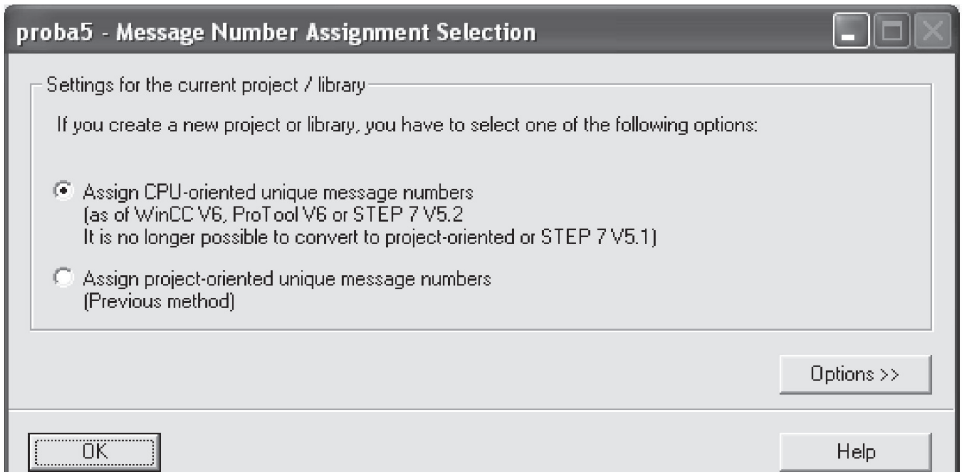


Рис. 3.3. Окно выбора назначения номеров сообщений

Теперь мы можем начинать работать над созданием мнемосхем в графическом редакторе. Для этого необходимо щелкнуть правой кнопкой мыши на *OS (1)* и выбрать команду *Open Object* (*рис. 3.4*).

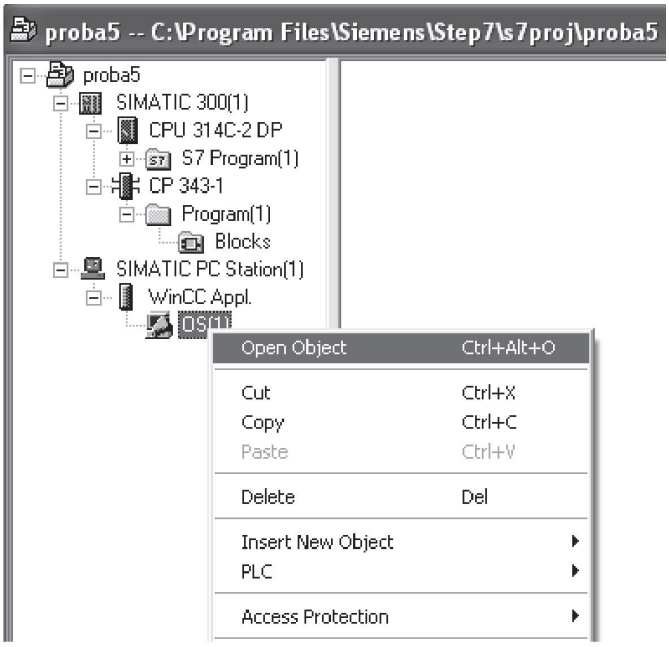


Рис. 3.4. Открытие проводника WinCC

Откроется окно проводника WinCC.

3.1. Создание мнемосхем

1. Заходим в проводнике WinCC в *Графический редактор* и создаем новый кадр мнемосхемы (*рис. 3.5*).

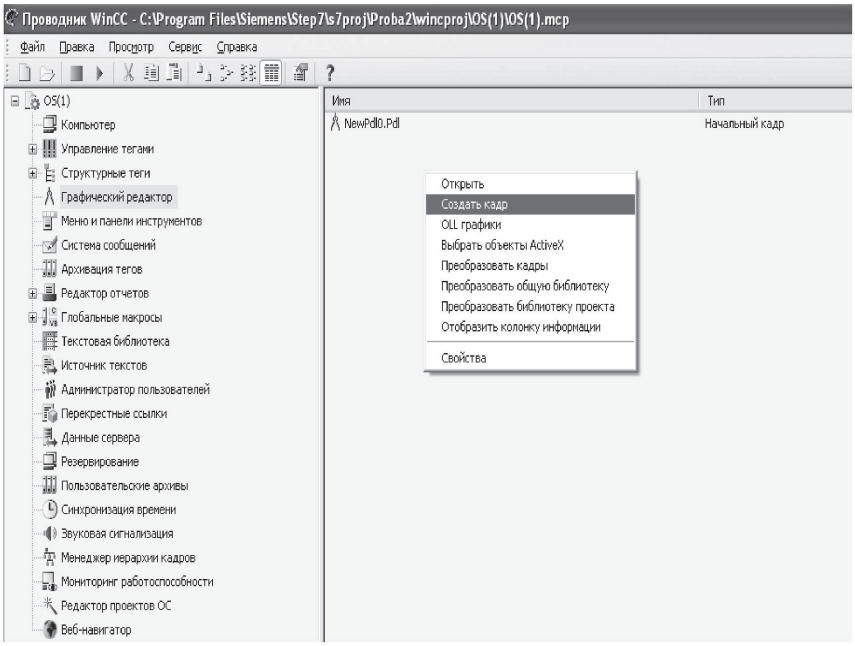


Рис. 3.5. Создание графического кадра

2. Кликаем правой клавишей по вновь созданному кадру и переименовываем его (*рис. 3.6*).

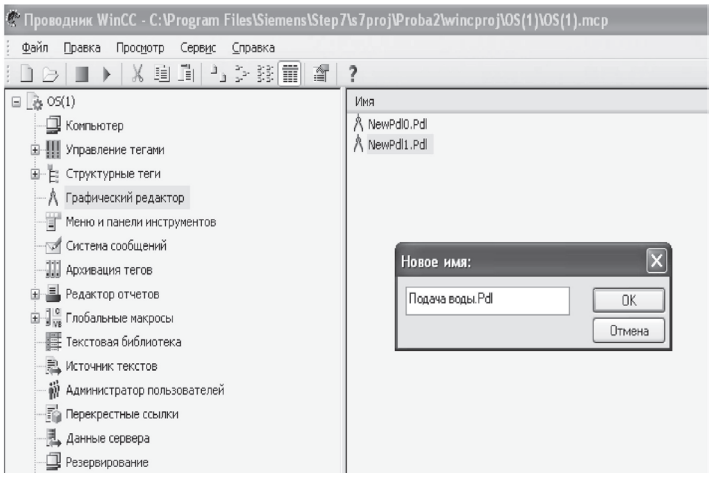


Рис. 3.6. Переименование графического кадра

3. Приступаем к созданию статических элементов, пользуясь библиотекой элементов (*рис. 3.7*).

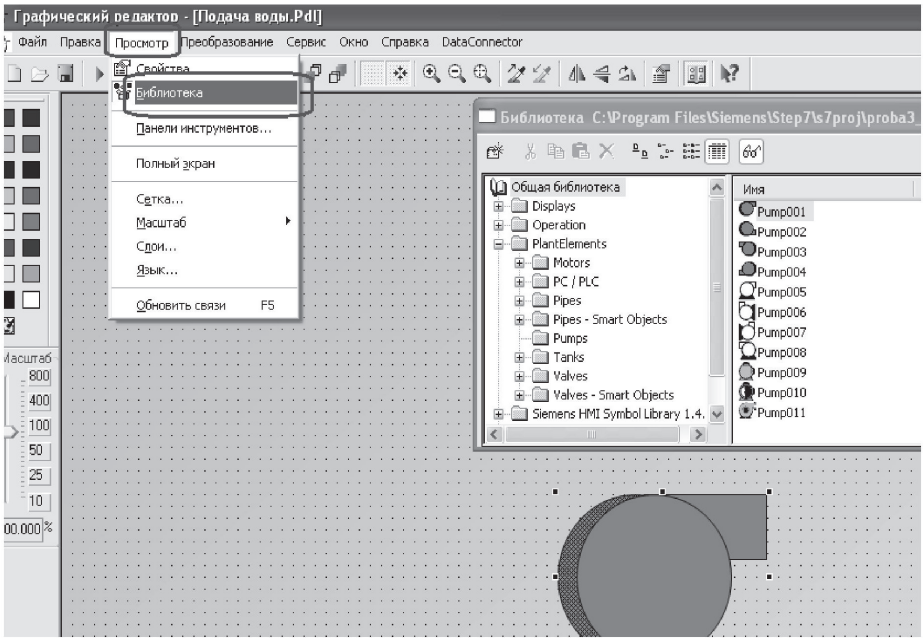


Рис. 3.7. Открытие библиотеки элементов

4. Когда все элементы созданы, на экране должна появиться такая картина (*рис. 3.8*).

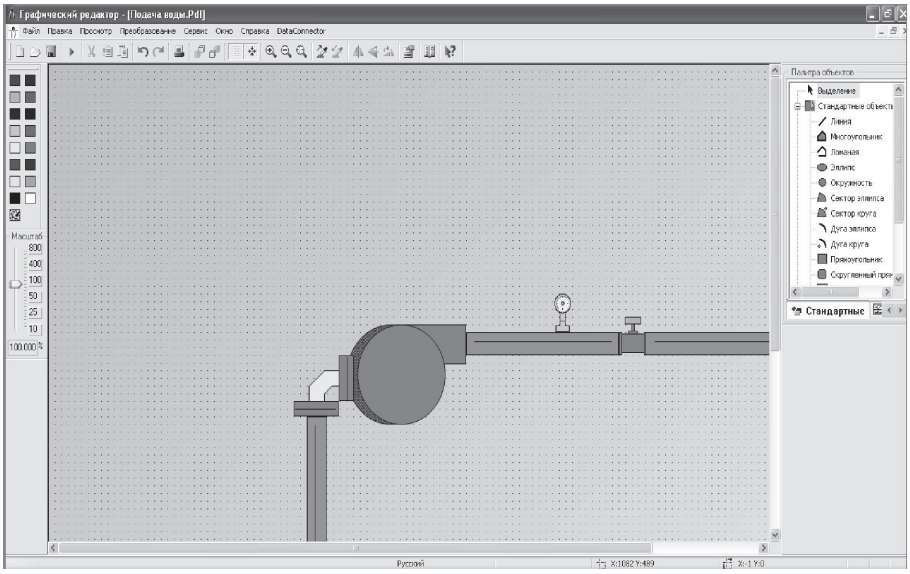


Рис. 3.8. Статические элементы на мнемосхеме

3.2. Работа с таблицей символов через Symbol Server

Теперь нам требуется привязать нашу мнемосхему к полевому оборудованию автоматизации, для этого следует создать динамические объекты. Начнем с указателя положения клапана. В палитре объектов выбираем раздел *Интеллектуальные объекты* и вытаскиваем на экран *Поле ввода-вывода* (*рис. 3.9*).

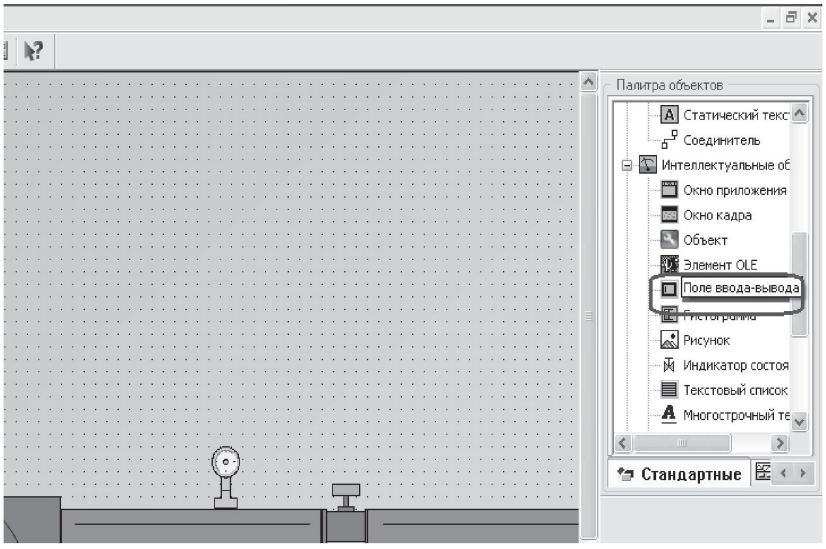


Рис. 3.9. Добавление динамических элементов на мнемосхему

Размещаем в нужном месте мнемосхемы — над клапаном — и приступаем к привязке тега (*рис. 3.10*).

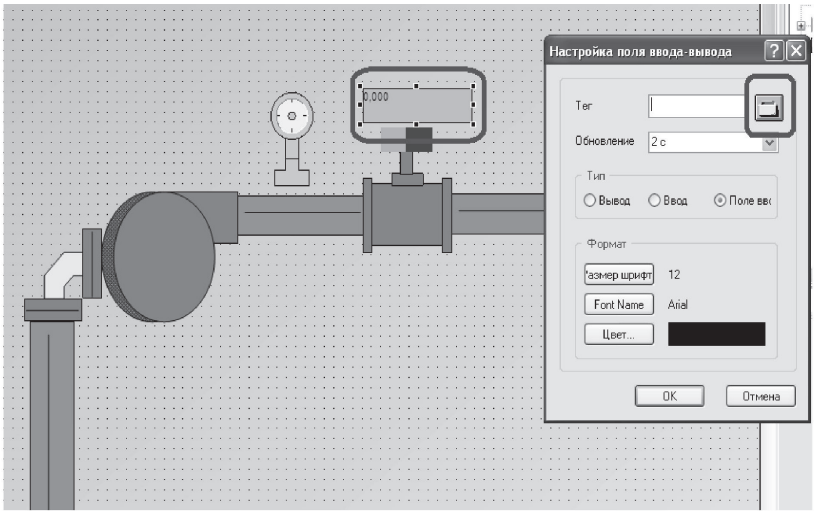


Рис. 3.10. Настройка поля ввода-вывода

Выбираем нужный тег из глобального блока данных, которому мы присвоили символьное имя Data_for_valve (*рис. 3.11*).

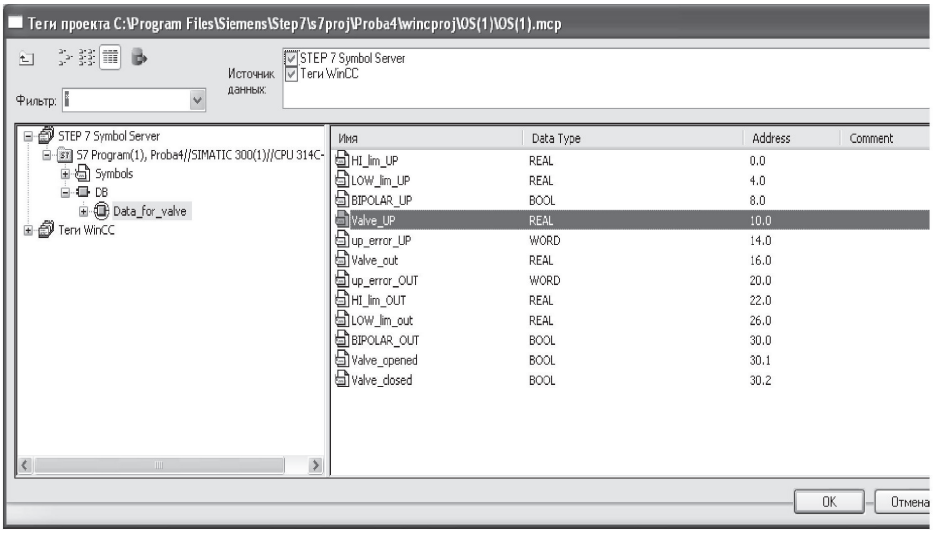


Рис. 3.11. Выбор нужного тега

При первом выборе тега из Step7 Symbol Server необходимо указать сеть, с помощью которой данные будут передаваться из контроллера на станцию оператора. Выбираем созданную нами сеть Ethernet (1) с протоколом TCP/IP и IP-адресом 192.168.0.5 (*рис. 3.12*).

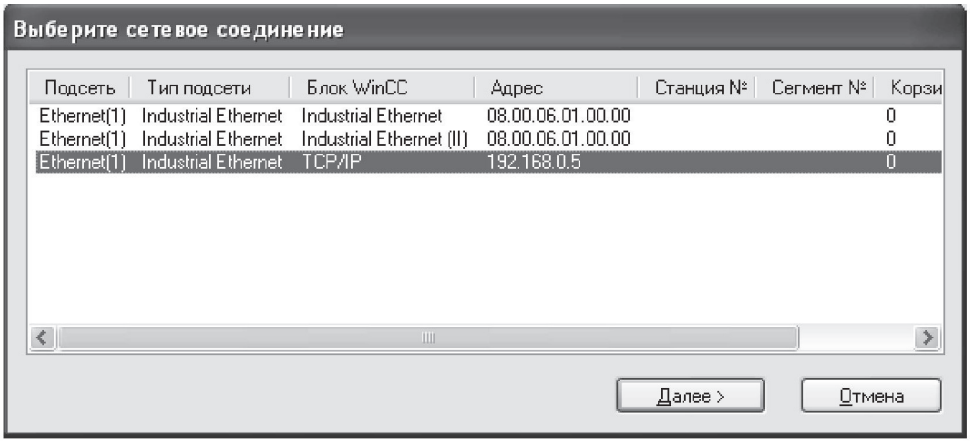


Рис. 3.12. Выбор сетевого соединения

Затем выбираем период обновления значения 250 мс, тип поля — вывод и нажимаем кнопку *OK*.

В результате получаем на мнемосхеме элемент отображения состояния оборудования. Если нажать по нему правой клавишей мыши и выбрать команду *Свойства*, то откроется окно, в котором можно настроить все свойства данного объекта.

3.3. Управление механизмами (создание выпадающих окон)

Для управления различными механизмами (запорно-регулирующей арматурой, насосами, вентиляторами и т. д.) требуется создать несколько управляющих объектов в зависимости от числа сигналов управления (например, для клапана: «указатель положения», «задание», «ползунок»). Чтобы не перегружать экран мнемосхемы этими элементами управления, применяется технология выпадающих окон, которые появляются при нажатии левой кнопки мыши на объект управления и исчезают при нажатии правой.

1. Создаем выпадающее окно при помощи инструментов *Прямоугольник*, *Статический текст*, *Поле ввода-вывода*, *Ползунок* из палитры объектов (*рис. 3.13*).

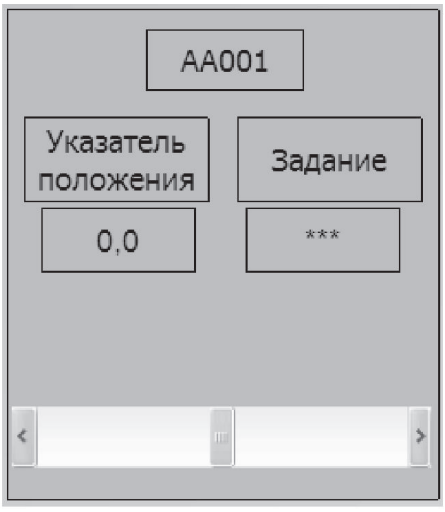


Рис. 3.13. Окно управления клапаном

2. Поле ввода-вывода, расположенное под надписью *Указатель положения*, привязываем к тегу Valve_UP и настраиваем данное поле только на вывод значения. Также в свойствах этого объекта можно

изменить шрифт, выравнивание, задать верхние и нижние границы, уменьшить количество знаков после запятой и т. д.

3. Поле ввода-вывода, расположенное под надписью *Задание*, и объект *Ползунок* привязываем к тегу Valve_OUT и также настраиваем свойства этих объектов.

4. Очень важно настроить верхние и нижние границы для данных объектов, поскольку наш клапан достигает крайних положений уже при значениях указателя положения 16 % при закрытии и 92 % при открытии. Данные значения и вводим в соответствующие поля.

5. Выделяем все объекты, удерживая клавишу *Shift* и группируем их (*рис. 3.14*).

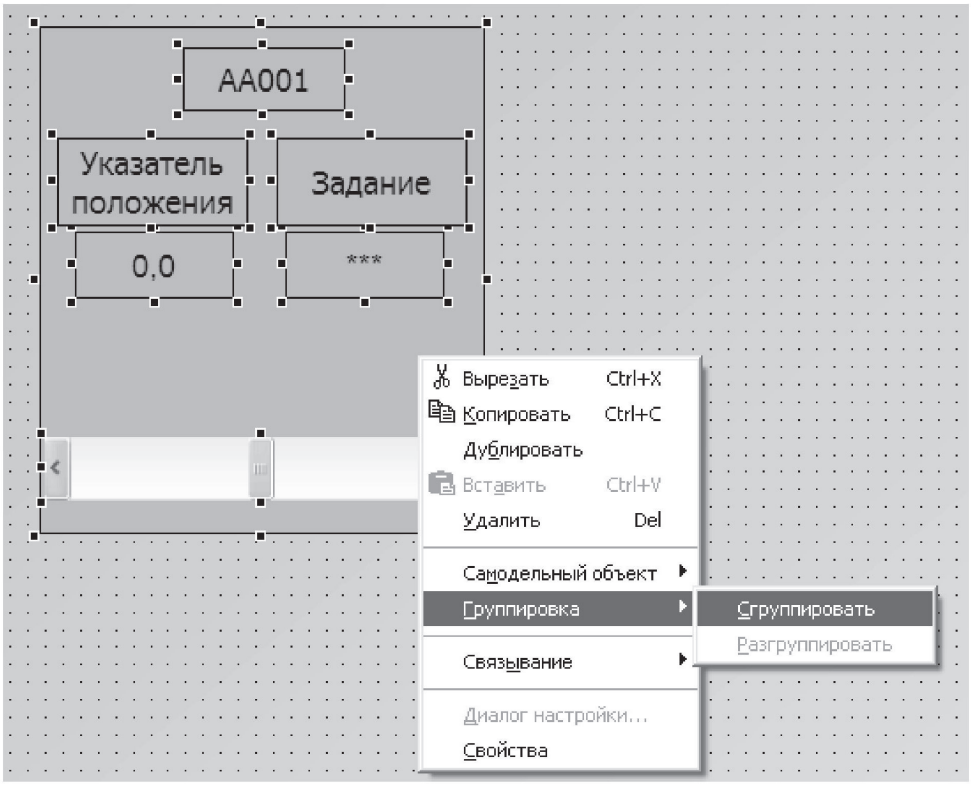


Рис. 3.14. Группирование объектов

6. Присваиваем созданному объекту имя управляемого клапана (например, AA001), *рис. 3.15.*

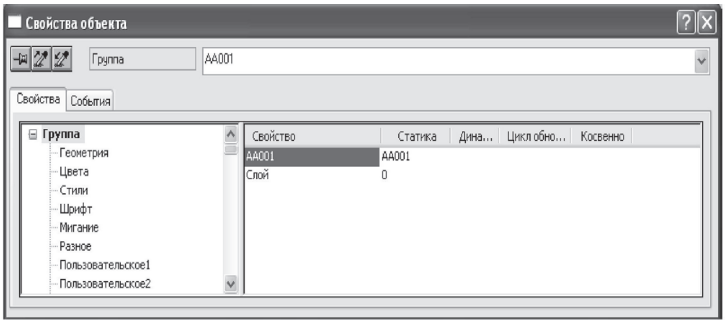


Рис. 3.15. Изменение имени группы объектов

7. В группе *Разное* присваиваем свойству *Отображать* значение *Нет*, теперь объект не будет отобажаться на экране (*рис. 3.16*).

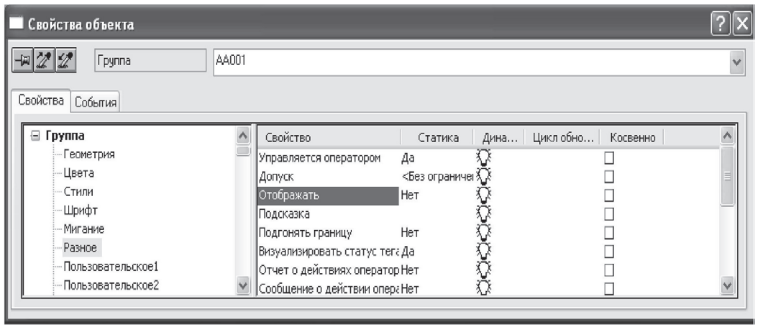


Рис. 3.16. Скрытие группы объектов на видеокадре в режиме Runtime

8. Теперь объединим все элементы иконки нашего клапана в группу и назовем ее AA001_ico (*рис. 3.17*).

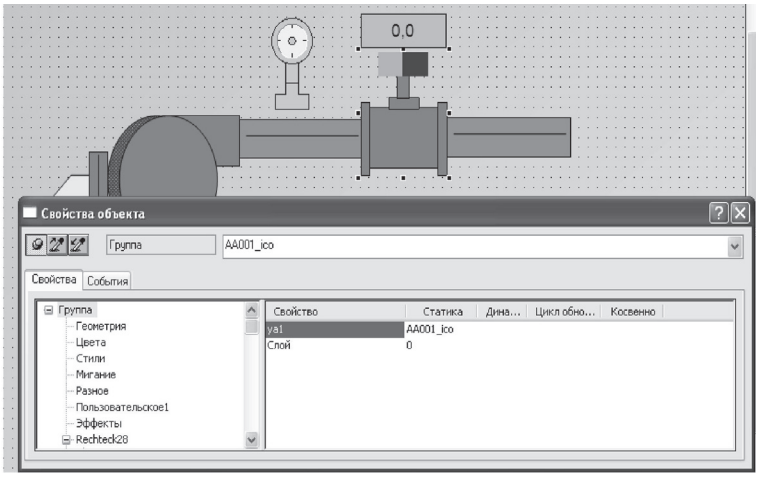


Рис. 3.17. Создание иконки клапана

9. Теперь нужно настроить события для данной иконки. При нажатии на нее левой кнопкой мыши должно открываться окно управления клапаном AA001, а при нажатии правой — закрываться.

10. Для этого открываем свойства и на вкладке *События* заходим в группу свойств *Мышь*, выбираем *Нажатие левой* и *Прямое соединение* (*рис. 3.18*).

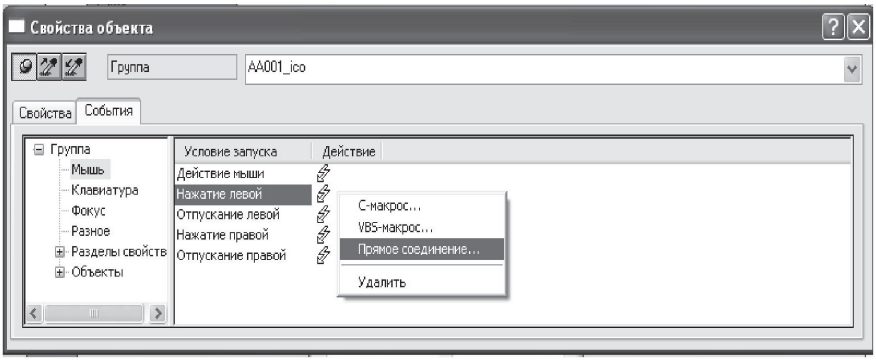


Рис. 3.18. Создание прямого соединения для события «Нажатие левой»

11. Далее настраиваем появившееся окно так, как показано ниже (*рис. 3.19*).

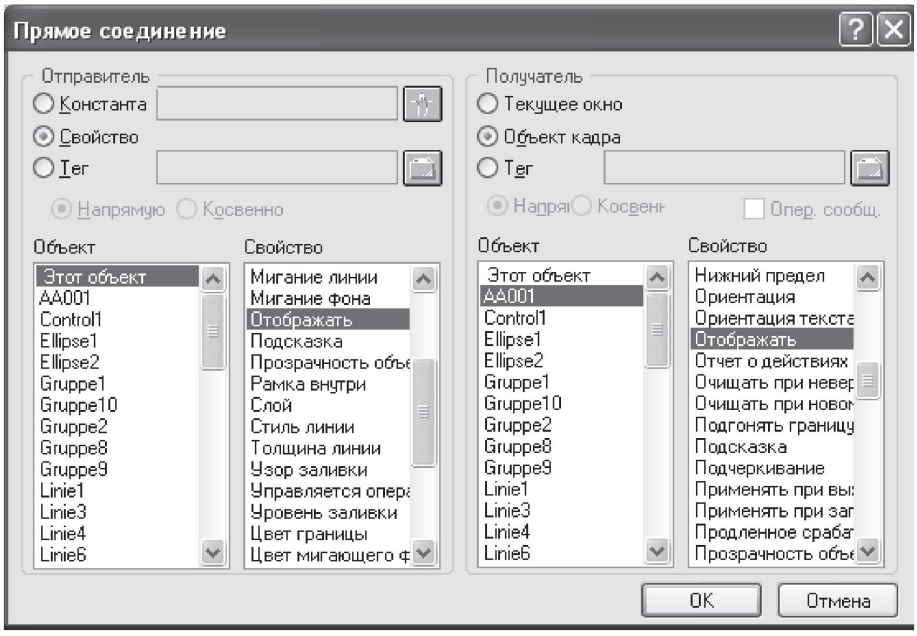


Рис. 3.19. Настройка прямого соединения для события «Нажатие левой»

Таким образом, при нажатии на иконку клапана настройка свойства отображения группы объектов AA001_ico скопируется в свойство отображения группы объектов AA001. То есть окно управления клапаном AA001 будет отображаться.

12. Аналогично настраиваем свойство мыши *Нажатие правой*. Только в поле *Отправитель* выбираем *Константа* и задаем значение 0. При нажатии правой кнопки мыши окно будет исчезать, поскольку в свойство отображения окна управления клапаном AA001 запишется значение 0 — не отображать (*рис. 3.20*).

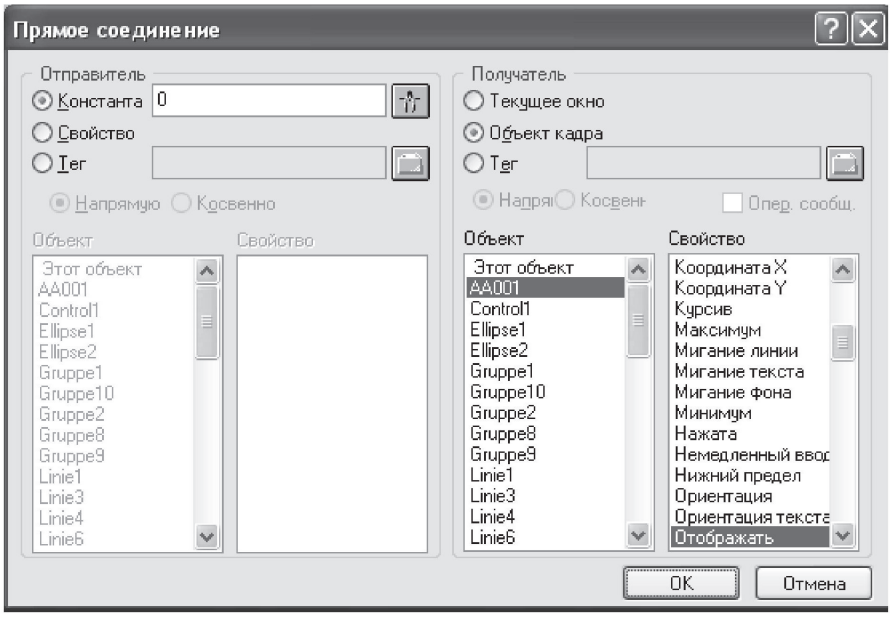


Рис. 3.20. Настройка прямого соединения для события «Нажатие правой»

13. Теперь можно включить режим выполнения *Runtine* и проверить отображение окна управления клапаном AA001 при нажатии на иконке клапана. Для этого необходимо нажать на кнопку *Среда исполнения* (*рис. 3.21*).

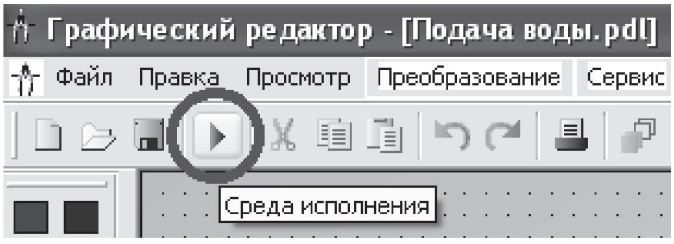


Рис. 3.21. Активация режима выполнения Runtime

14. Поскольку станция оператора не подключена к контроллеру, то в полях интеллектуальных объектов отображаются значки в виде восклицательного знака в желтом треугольнике, говорящие об отсутствии связи с контроллером (*рис. 3.22*).

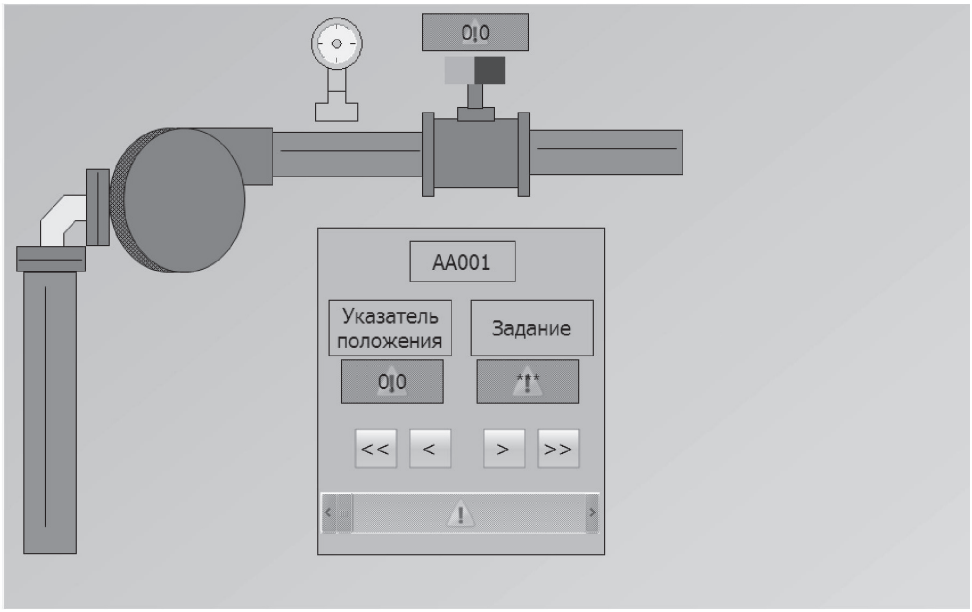


Рис. 3.22. Режим выполнения Runtime

15. Здесь мы рассмотрели создание выпадающих окон для выработки управляющих воздействий. Для того чтобы отображать концевые положения клапана (*Открыто*, *Закрыто*), можно также создать элементы отображения в выпадающем окне, но лучше ввести цветовую идентификацию, когда графическое изображение задвижки меняет свой цвет в зависимости от сигнала. Для этого привяжем свойство *Цвет фона* прямоугольников, расположенных над клапаном, к значениям концевых выключателей клапана. При наличии сигнала концевика *Закрыто* правый красный прямоугольник должен становиться зеленым. А при наличии сигнала концевика *Открыто* левый зеленый прямоугольник должен становиться красным (*рис. 3.23* и *3.24*).

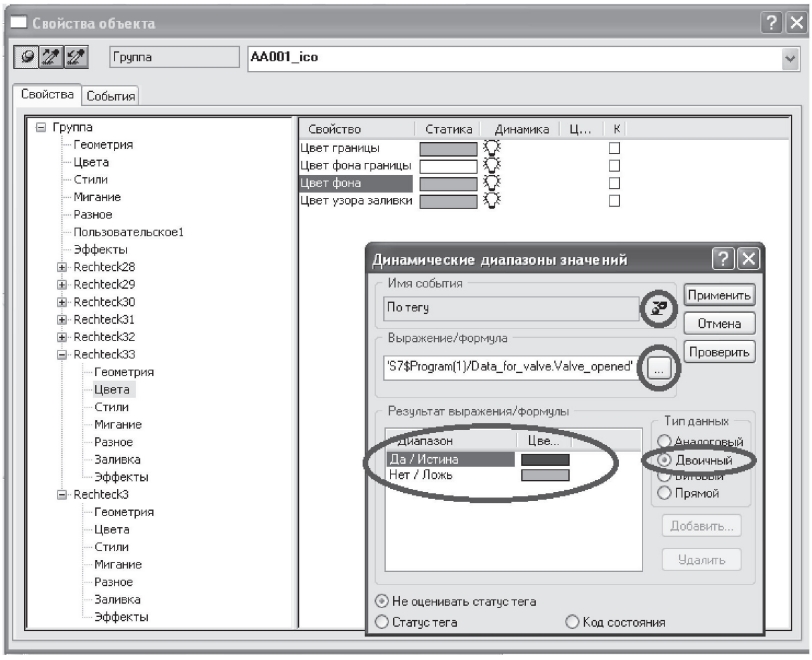


Рис. 3.23. Подключения концевика «Открыто»

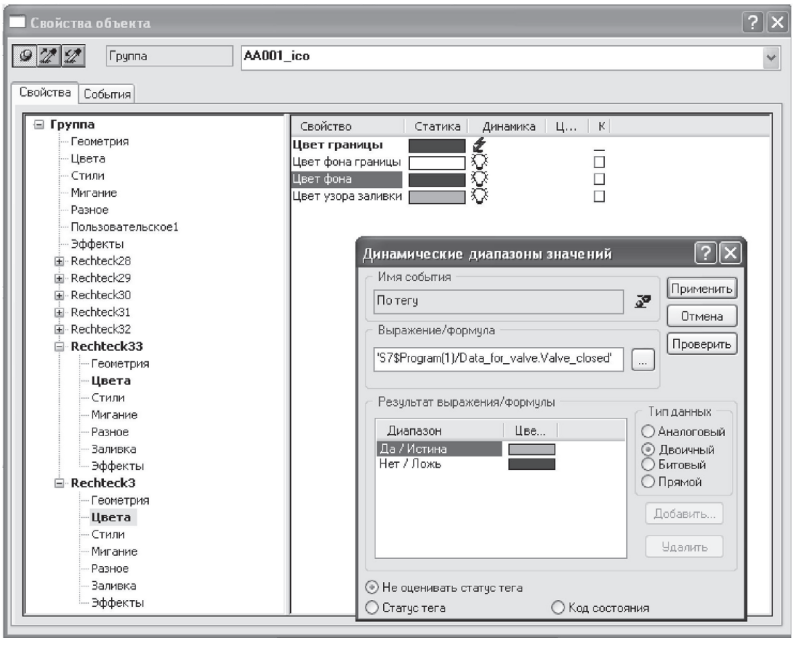


Рис. 3.24. Подключения концевика «Закрыто»

Для начала выбираем тег, затем настраиваем период обновления состояния с помощью кнопки . После этого выбираем тип данных и затем создаем соответствие между диапазоном и цветом прямоугольника.

3.4. Отладка проекта

Далее нам надо запустить обе программы (Step7 и WinCC) в режиме симуляции, в WinCC это делается при помощи нажатия на кнопку *Запустить* (*рис. 3.25*).

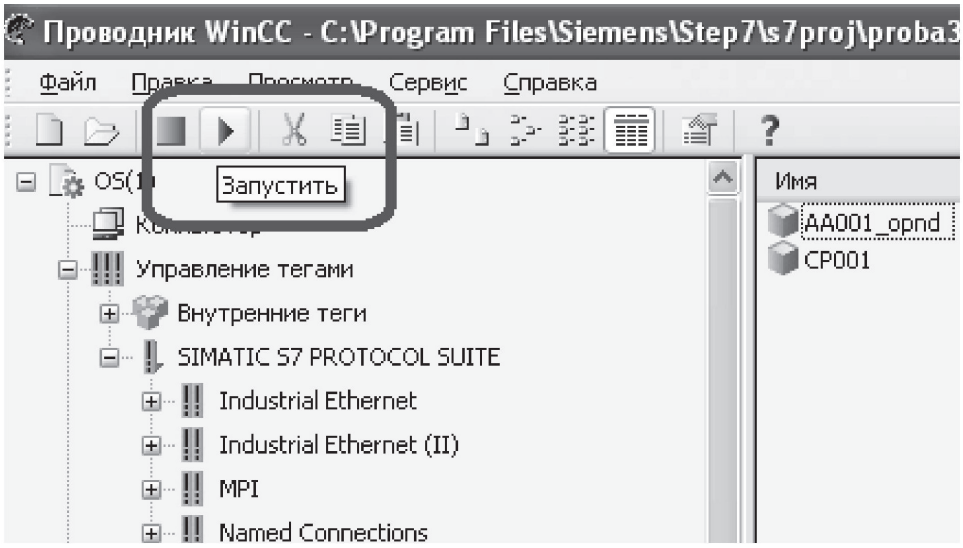


Рис. 3.25. Активация режима выполнения «Runtime»

На экране должен появиться начальный экран процесса в режиме WinCC Runtime.

После чего запускаем программу в Step7 в режиме симулятора.

Кроме того, необходимо переключить соединение станции оператора с контроллером на соединение ее со симулятором для передачи данных с помощью Symbol Server. Для этого в проводнике WinCC выбираем *Управление тегами*, *SIMATIC S7 PROTOCOL SUITE*, нажимаем правой кнопкой на *TCP/IP* и выбираем *Системные параметры* (*рис. 3.26*).

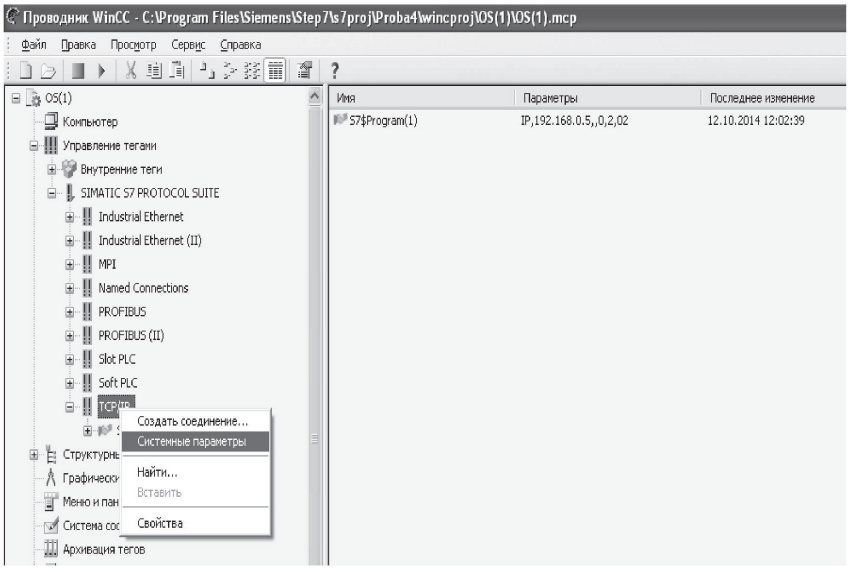


Рис. 3.26. Изменение настроек в соединении для Symbol Server

Выбираем вкладку *Устройство* и меняем логическое имя на PLCSIM (TCP/IP), нажимаем кнопку *OK*. Будет предложено закрыть наш проект, соглашаемся с этим и затем открываем проект вновь (*рис. 3.27*).

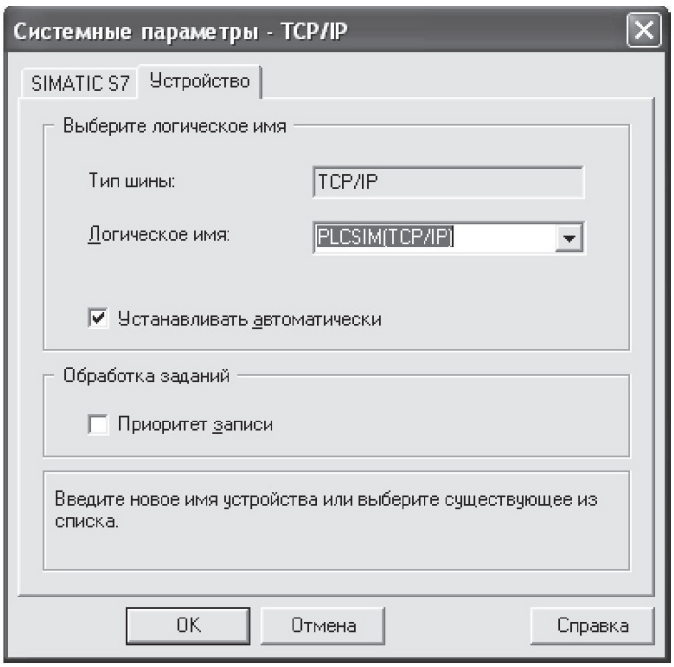


Рис. 3.27. Выбор соединения PLCSIM (TCP/IP) для Symbol Server

Если все было сделано правильно, то ваши действия в Step7 будут тут же отображаться на экране WinCC (*рис. 3.28*).

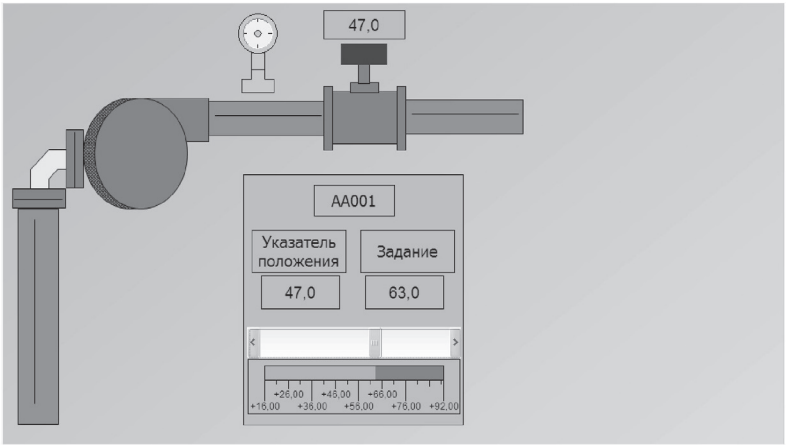


Рис. 3.28. Мнемосхема в режиме выполнения Runtime

Как мы видим (*рис. 3.29*), значения в блоке данных DB2 и на мнемосхеме совпадают, и сигнал концевого выключателя также работает правильно.

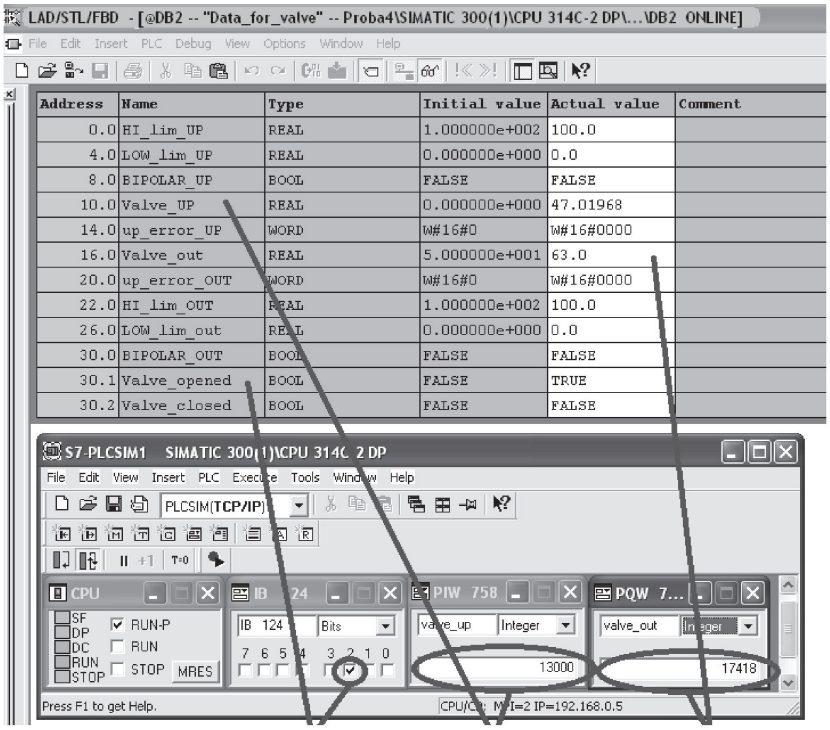


Рис. 3.29. Задание входных сигналов в симуляторе и наблюдение за значениями в блоке данных

После того как проверили работу программы в симуляторе, можно подключать контроллер. Но перед этим нужно настроить сетевое соединение для связи с контроллером.

Теперь можно подключить контроллер и проверить работу нашей программы на реальном оборудовании (*рис. 3.30*, *3.31*).

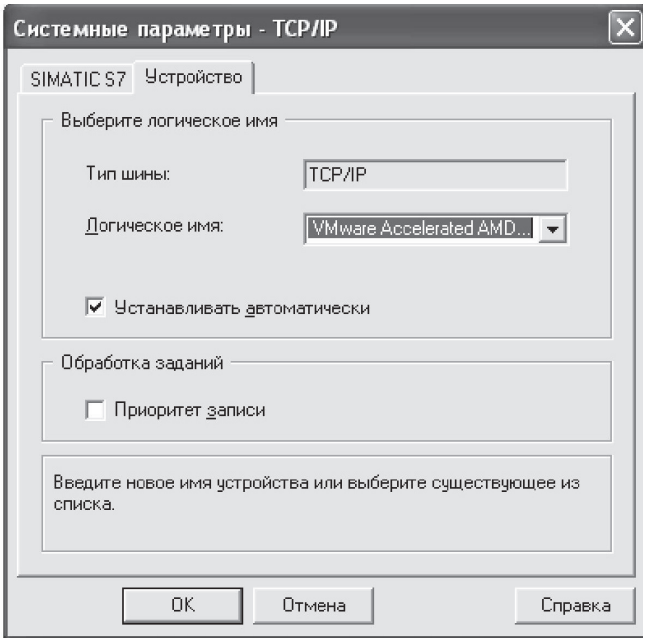


Рис. 3.30. Настройка связи WinCC и контроллера

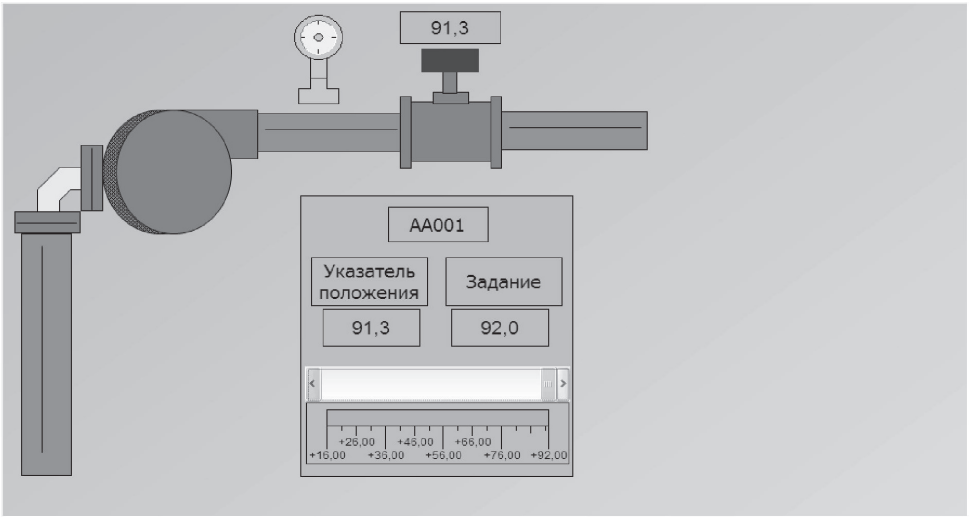


Рис. 3.31. Управление реальным клапаном

Из данного окна можно управлять клапаном с помощью окна ввода-вывода, а также с помощью ползунка. Для удобства рядом с ползунком была размещена гистограмма. Величина задания отображается в поле *Задание* и на гистограмме. Концевые выключатели отображаются на иконке клапана. Величина указателя положения выводится на поля вывода над клапаном и в окне управления.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Siemens SIMATIC. WinCC V7.0. Начало работы. Печатная версия интерактивной справки. — Берлин : Изд-во Siemens AG, 2008. — 254 с. [Электронный ресурс]. — URL: http://dfpd.siemens.ru/assets/files/infocenter/Documetations/Automation_systems/HMI/WinCC/V70/GettingStarted_ru.pdf (06.07.2016).
2. Бергер Г. Автоматизация с помощью программ STEP7 LAD и FBD. — 2-е изд., перераб. / Г. Бергер. — Нюрнберг : Siemens AG, 2001. — 605 с. [Электронный ресурс]. — URL: http://dfpd.siemens.ru/assets/files/infocenter/Documetations/Automation_systems/STEP7/Berger_STEP7_LAD&FBD_r.pdf (06.07.2016).

Учебное издание

Кисельников Андрей Юрьевич,
Худяков Павел Юрьевич,
Жеребчиков Алексей Юрьевич

ПРОГРАММИРОВАНИЕ ПТК SIEMENS И ПТК VIPA В ПРОГРАММНЫХ ПАКЕТАХ STEP7, WINCC И PCS7

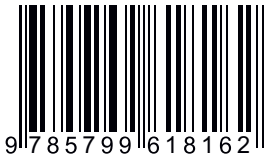
Редактор *О. В. Климова*
Корректор *Е. Е. Афанасьева*
Верстка *Е. В. Ровнушкиной*

Подписано в печать 19.08.2016. Формат 70×100 1/16.
Бумага писчая. Цифровая печать. Усл. печ. л. 6,8.
Уч.-изд. л. 3,5. Тираж 70 экз. Заказ 307.

Издательство Уральского университета
Редакционно-издательский отдел ИПЦ УрФУ
620049, Екатеринбург, ул. С. Ковалевской, 5
Тел.: 8 (343) 375-48-25, 375-46-85, 374-19-41
E-mail: rio@urfu.ru

Отпечатано в Издательско-полиграфическом центре УрФУ
620075, Екатеринбург, ул. Тургенева, 4
Тел.: 8 (343) 350-56-64, 350-90-13
Факс: 8 (343) 358-93-06
E-mail: press-urfu@mail.ru

Для заметок



КИСЕЛЬНИКОВ АНДРЕЙ ЮРЬЕВИЧ

Кандидат технических наук
Начальник цеха тепловой автоматики и измерений на ТЭЦ «Академическая» Свердловского филиала ПАО «Т Плюс» (г. Екатеринбург), доцент кафедры ТЭС Уральского энергетического института ФГАОУ ВПО «Уральский федеральный университет имени первого Президента России Б. Н. Ельцина».

ХУДЯКОВ ПАВЕЛ ЮРЬЕВИЧ

Кандидат физико-математических наук
Заведующий кафедрой автоматизации технологических процессов и производств НЧОУ ВО «Технический университет УГМК» (г. Верхняя Пышма), доцент кафедры ТЭС Уральского энергетического института ФГАОУ ВПО «Уральский федеральный университет имени первого Президента России Б. Н. Ельцина».

ЖЕРЕБЧИКОВ АЛЕКСЕЙ ЮРЬЕВИЧ

Ведущий инженер ООО «Уралэнергоналадка» (г. Екатеринбург).
Участвовал в проектах автоматизации более 10 энергоблоков ТЭС, в том числе 3 энергоблоков в Индии.